

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
ESCUELA DE POSGRADO



RESOLUCIÓN DE LA ECUACIÓN DE ADVECCIÓN LINEAL  
UNIDIMENSIONAL POR UN MÉTODO DE VOLUMENES  
FINITOS COMPACTO DE ALTO ORDEN

Tesis para optar el grado de Magíster en Matemáticas Aplicadas  
con mención en Procesos Estocásticos

AUTOR

Xyoby Chávez Pacheco

ASESOR

Dr. Juan Eduardo Casavilca Silva

JURADO

Dr. Rubén Angel Agapito Ruiz

Dra. Loretta Betzabe Rosa Gasco Campos

LIMA-PERÚ  
2017



*A mi esposa y mis hijos.*



# Agradecimientos

A Dios por darme la oportunidad de completar esta fase en el desarrollo de mi formación profesional. A mi familia por su cariño, comprensión y apoyo al aceptar el tiempo dedicado a este trabajo. A mi asesor el Dr Juan Casavilca por su paciencia, proximidad y acertadas observaciones. A la Universidad de Ingeniería y Tecnología UTEC por su constante respaldo; y a todas aquellas personas que de algún modo se vieron involucradas en la elaboración de esta tesis.

# Índice general

Lista de figuras	VI
Lista de Cuadros	IX
Abstract	XI
Resumen	XII
<b>1. Introducción</b>	<b>1</b>
<b>2. Métodos de Volúmenes Finitos (MVF )</b>	<b>3</b>
2.1. Forma integral de la ecuación de advección unidimensional. . . . .	3
2.2. Discretización espacial de la forma integral de la ecuación de ad- vección 1D. . . . .	4
2.3. Discretización temporal . . . . .	6
<b>3. MVF Compacto: Reconstrucción</b>	<b>10</b>
3.1. Reconstrucción de mínimos cuadrados compacto . . . . .	10
3.1.1. Condiciones de frontera de Dirichlet . . . . .	16
3.1.2. Condiciones de frontera periódicas . . . . .	17
3.1.3. Consideraciones de la reconstrucción . . . . .	19
3.2. Matrices de reconstrucción explícitas. . . . .	20
3.3. Validación numérica . . . . .	22
<b>4. MVF Compacto: Flujo</b>	<b>26</b>
4.1. Características y la solución general . . . . .	27

4.2. Elección del flujo numérico . . . . .	28
4.3. Esquema de volúmenes finitos para la ecuación de advección lineal . . . . .	31
<b>5. Análisis de Fourier</b>	<b>33</b>
5.1. Análisis espectral . . . . .	33
5.2. Cálculo de los números de onda modificados . . . . .	36
5.2.1. Número de onda modificado para un esquema de segundo orden ( $k = 1$ ) . . . . .	36
5.2.2. Número de onda modificado para esquemas de órdenes su- periores . . . . .	40
5.3. Optimización de parámetros . . . . .	41
5.4. Análisis de estabilidad de von Neumann . . . . .	43
5.4.1. Análisis con Runge-Kutta 3 . . . . .	44
5.4.2. Análisis con Runge-Kutta 4 . . . . .	44
5.4.3. Consideraciones del factor de amplificación . . . . .	46
<b>6. Validación del método</b>	<b>49</b>
6.1. Requerimientos de software y hardware . . . . .	50
6.2. Implementación y Algoritmo . . . . .	50
6.3. Resultados numéricos . . . . .	51
6.3.1. Condiciones de frontera de Dirichlet. . . . .	53
6.3.2. Condiciones de frontera periódicas . . . . .	57
<b>7. Conclusiones y trabajos futuros</b>	<b>63</b>
<b>A. Anexo</b>	<b>65</b>
A.1. Definiciones . . . . .	65
A.2. El Problema de Riemman . . . . .	67
A.2.1. Ecuación de Euler unidimensional . . . . .	67
A.2.2. Sistemas de ecuaciones lineales hiperbólicos . . . . .	69
A.2.3. El Problema de Riemman para un sistema de ecuaciones hiperbólicos. . . . .	71
A.3. Ley de conservación . . . . .	73
A.3.1. La forma integral de la ley de conservación . . . . .	73

B. Subprogramas	75
Bibliografía	82



# Índice de figuras

2.1. Valores medios $\bar{u}_i$ de la función $u(t, x)$ en cada intervalo $I_i$ para un tiempo fijo $t$ . . . . .	5
2.2. Evolución en el tiempo de los valores medio $\bar{u}_i$ . . . . .	7
3.1. Tratamiento de las condiciones de frontera periódicas . . . . .	18
3.2. Reconstrucción lineal y valores promedio de $u(x) = \cos(\pi x)$ en $[-1, 1]$ con $N = 4$ (a) $N = 8$ (b) y $N = 16$ (c) intervalos . . . . .	23
3.3. Reconstrucción cuadrática y valores promedio de $u(x) = \cos(\pi x)$ en $[-1, 1]$ con $N = 4$ (a) $N = 8$ (b) y $N = 16$ (c) intervalos . . . . .	24
3.4. Curvas de convergencia de primer a sexto orden para una reconstrucción constante por partes ( $k = 0$ ), lineal ( $k = 1$ ) , parabólico ( $k = 2$ ), cúbico ( $k = 3$ ), $k = 4$ y $k = 5$ para la función $u(x) = \cos(\pi x)$ en $[-1, 1]$ con parámetros $w_i = 1$ . . . . .	24
4.1. Curvas características de la ecuación de advección lineal para una velocidad característica positiva $a$ . Condición inicial en el tiempo $t = 0$ fija la posición inicial en $x_0$ . . . . .	27
4.2. Condición inicial para un problema de Riemman. En el tiempo inicial el dato consiste de dos constantes de estado separadas por una discontinuidad en $x = 0$ . . . . .	29
4.3. Ilustración de la solución del problema de Riemman en el plano $x-t$ para la ecuación de advección lineal con velocidad de característica positiva $a$ . . . . .	29
4.4. Elección de flujo entre dos polinomios de reconstrucción $u_i(x)$ y $u_{i+1}(x)$ . . . . .	30



5.1. Longitud de una celda $I_i$ . . . . .	37
5.2. Propiedades de dispersión (a) y disipación (b) de esquemas CLSFV de segundo a cuarto orden . . . . .	42
5.3. Error de dispersión para esquemas CLSFV de segundo a cuarto orden con un ancho de banda del 3%. . . . .	43
5.4. Factor de amplificación para los esquemas CLSFV usando TVD RK3. Los números CLF son 0.6, 0.8, 1.0, 1.1, 1.2 y 1.4 en (a) -(f) respectivamente. . . . .	47
5.5. Factor de amplificación para los esquemas CLSFV usando RK4. Los números CFL son 0.6, 0.8, 1.0, 1.1, 1.2 y 1.4 en (a) -(f) respectivamente. . . . .	48
6.1. Solución de la ecuación advección (2.2) con condiciones de frontera Dirichlet y condición inicial $u_0(x) = \text{sen}(2\pi x)$ . La solución exacta es $u(x, t) = u_0(x - t)$ en $[-1, 1]$ y se muestra las aproximaciones con reconstrucciones lineales para $N = 4$ (a) $N = 8$ (b) y $N = 32$ (c) usando esquemas de Runge-Kutta clásico de orden cuatro para la evolución temporal. . . . .	54
6.2. Solución de la ecuación advección (2.2) con condiciones de frontera periódicos y condición inicial $u_0(x) = \text{sen}(2\pi x)$ . La solución exacta es $u(x, t) = u_0(x - t)$ en $[-1, 1]$ y se muestra las aproximaciones con reconstrucciones lineales para $N = 4$ (a) $N = 8$ (b) y $N = 32$ (c) usando esquemas de Runge-Kutta clásico de orden cuatro para la evolución temporal. . . . .	57
6.3. Error de aproximación para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción orden cuatro ( $k = 3$ ) con parámetros no óptimos $w_1 = 1 = w_2$ y óptimos $w_1 = 0,01 = w_2$ , RK4 y CFL=1 . . . . .	60
6.4. Orden de convergencia para solución de la ecuación de advección unidimensional para diferentes órdenes con parámetros óptimos cuando $k = 2, 3$ ; $w_i = 1,0$ para $k = 5, 6$ ; con condiciones de frontera periódicas, CFL=1 y RK4. . . . .	62

A.1. Estructura de la solución del problema de Riemman para un sistema lineal hiperbólico general $m \times m$ con coeficientes constantes. . . . .	72
A.2. Volumen de control $V = [x_L, x_R] \times [t_1, t_2]$ sobre el plano $x - t$ . . . . .	74



# Índice de cuadros

3.1. Errores de aproximación de la función $u(x) = \cos(\pi x)$ para la reconstrucción de orden alto y sus respectivos órdenes numéricos, con $\tilde{u}(x)$ la función aproximada. . . . .	25
5.1. Valores optimizados de $(w_1, w_2)$ para diferentes valores de $\nu$ (Tomado de [22]) . . . . .	42
6.1. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet para orden 4 ( $k = 3$ ) con parámetros no optimizados $w_1 = 1 = w_2$ , y optimizados $w_1 = 0,01$ , $w_2 = 0,01$ ; $w_1 = 1, w_2 \approx 0$ ; con RK3 y $CFL = 1$ . . . . .	55
6.2. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet, RK3 y $CFL = 1$ . . . . .	55
6.3. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet para tercer ( $k = 2$ ) y cuarto orden ( $k = 3$ ) con parámetros $w_1 = 0,00001 \approx 0$ ; $w_1 = 1 = w_2$ y $w_1 = 0,01 = w_2$ , RK4 y $CFL=1$ . . . . .	56
6.4. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet con RK4 y $CFL=1$ . . . . .	56

6.5. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos, para esquemas de reconstrucción de cuarto orden ( $k = 3$ ) con parámetros no optimizados $w_1 = 1, w_2 = 1$ y optimizados $w_1 = 0,01, w_2 = 0,01$ ; $w_1 = 1; w_2 = 0$ ; con RK3 y CFL=1 . . . . .	58
6.6. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción de diferentes órdenes con RK3 y CFL=1 . . . . .	59
6.7. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción orden cuarto ( $k = 3$ ) con parámetros no óptimos $w_1 = 1 = w_2$ y óptimos $w_1 = 0,01 = w_2$ , RK4 y CFL=1 . . . . .	60
6.8. Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción con diferentes órdenes, con parámetros óptimos cuando $k = 2, 3$ y $w_i = 1,0$ para $k = 5, 6$ ; usando RK4 y CFL=1 . . . . .	61

# Abstract

The numerical methods of high order, necessary for spatial discretization, are one of the most active areas of the field of Computational Fluid Dynamics. Within these, Finite Volume Methods (abbreviated as MVF in spanish) have encountered difficulties in the implementation of reconstruction processes. In the present work we present a novel high order compact reconstruction process proposed by Q. Wang [22], and implemented in Python. The novelty lies in that high order is achieved using a compact stencil, that is, using only neighboring cells. In this process we obtain a set of relations that are constructed to obtain the coefficients of reconstruction polynomials on the control volumes of interest, preserving their average values and that of their derivatives. With these relations we obtain an overdetermined linear system that is adjusted by least squares resulting in a tridiagonal system by blocks in the case of a 1D advection equation. For this advection equation we also use the Fourier Analysis to examine the wave numbers modified by the compact MVF. The reconstruction includes parameters that are optimized to improve the dispersion / dissipation properties. Furthermore, the von Neumann stability analysis allows us to estimate the maximum CFL number for two Runge-Kutta methods. Finally, we validate the convergence orders of the combination of the compact MVF with two schemes of Runge-Kutta and we also validate the optimal parameters of the reconstruction schemes.

# Resumen

Los métodos numéricos de alto orden, necesarios para la discretización espacial, son una de las áreas más activas del campo de la dinámica de fluidos computacional (CFD en sus siglas en inglés). Dentro de estos, los Métodos de Volúmenes Finitos (MVF) han encontrado dificultades en la implementación de los procesos de reconstrucción. En el presente trabajo presentamos e implementamos en Python un novedoso proceso de reconstrucción compacto de alto orden propuesto por Q. Wang [22]. La novedad yace en que el orden alto es alcanzado usando un estencil compacto; es decir, usando únicamente celdas vecinas. En este proceso se obtiene un conjunto de relaciones que sirven para obtener los coeficientes de los polinomios de reconstrucción sobre los volúmenes de control de interés preservando sus valores promedios y el de sus derivadas. Con estas relaciones obtenemos un sistema lineal sobredeterminado que al ajustarse por mínimos cuadrados resultan en un sistema tridiagonal por bloques para el caso de una ecuación de advección 1D. Para esta ecuación de advección usamos además el Análisis de Fourier para examinar los números de onda modificados por el MVF compacto. La reconstrucción incluye parámetros que son optimizados para mejorar las propiedades de dispersión/disipación. Así mismo, el análisis de estabilidad de von Neumann nos permite estimar el número CFL (Courant Friedrich Levy) máximo para dos métodos de Runge-Kutta. Finalmente, validamos tanto los órdenes de convergencia de la combinación del MVF compacto con dos esquemas de Runge-Kutta como los parámetros óptimos de los esquemas de reconstrucción.

# Capítulo 1

## Introducción

Blazek [2] sostiene que la Dinámica de Fluidos Computacional (CFD Computational Fluid Dynamics sus siglas en inglés) es uno de los campos más emergentes de la mecánica de fluidos. Los modelos matemáticos que los representan usan métodos numéricos para resolverlas, pero como complementa Blade[1], estos métodos numéricos producen oscilaciones en la solución en zonas cercanas a grandes cambios, en especial en zonas cercanas a discontinuidades de la solución. Las oscilaciones pueden ir aumentando al transcurrir el tiempo y provocar que el esquema numérico sea inestable o simplemente que aproxime mal a la realidad. Este fenómeno se conoce también por el nombre de dispersión. Pudiera ocurrir también que estos métodos sean disipativos; es decir, que la amplitud de una onda decrezca más que su solución exacta, llevando a que las discontinuidades en las regiones de alta variación, sean esparcidas por el dominio de la solución.

Dentro de estos métodos numéricos se encuentran aquellos denominados de alto orden, que para Wang[22] tienen la ventaja de tener aproximaciones de alto orden, baja disipación, baja dispersión y alta capacidad para manejar geometrías complejas.

En las últimas décadas se han desarrollado diferentes métodos de alto orden, como los métodos de volúmenes finitos k-exacto, los MVF de tipo esencialmente no oscilatorios (ENO) [12], los MVF ENO ponderados (WENO) [17], métodos de Galerkin continuo (CG) y discontinuo (DG) [14],[13], volumen espectral (SV) [25], entre otros.

Los MVF pueden alcanzar altos órdenes sobre mallas no estructuradas. Olliver-Gooch [11] y Versteeg [20], Blazek [2] y Moukalled [10] nos dan una idea del desarrollo de estos métodos aplicados al CFD.

En base a los trabajos de Wang[23, 24] y Zhang[27], Wang en [22] asegura que uno de los mayores problemas que afrontan los MVF de alto orden es la cardinalidad del conjunto estencil, conformado por elementos o celdas vecinas usadas en la discretización para asegurar un determinado orden. Este problema ocasiona: primero, la pérdida de memoria cache; segundo, el deterioro de la eficiencia en el paralelismo; y tercero, la necesidad de más memoria para almacenar información asociada.

En el presente trabajo estudiamos un proceso novedoso de reconstrucción compacto para los MVF de alto orden propuesto por Q. Wang [22] y lo implementamos en el lenguaje de programación Python. La novedad del proceso yace en el hecho que el orden alto es alcanzado usando un estencil compacto, esto es, usando únicamente las celdas vecinas más próximas. En este proceso se consigue un conjunto de relaciones, cuyo fin es obtener polinomios de reconstrucción, de modo que preserven el valor promedio de sus derivadas sobre los volúmenes de control local y vecinos. Se consigue así un sistema de ecuaciones lineales sobredeterminado que es resuelto usando la técnica de mínimos cuadrados.

Bajo estos fundamentos hemos organizado esta tesis como sigue:

En el Capítulo 2, presentamos una breve introducción a los MVF; seguidamente en el Capítulo 3 estudiamos en detalle el esquema de reconstrucción compacta propuesto por Wang [22]. En el Capítulo 4, se estudia el otro elemento importante en los MVF, el cálculo de flujo numérico. Como ejemplo se estudian las soluciones características de una ecuación diferencial parcial de primer orden de tipo advectiva unidimensional y cómo ésta determina la elección del flujo numérico.

En el Capítulo 5, abordamos el estudio espectral del método poniendo énfasis en el análisis de Fourier para estudiar las propiedades de disipación, dispersión y de estabilidad. Esta sección es la más significativa en términos de los recursos usados en la implementación bajo la programación simbólica.

En el Capítulo 6, se muestran los resultados numéricos obtenidos después de implementarlos en Python y finalmente, en el último capítulo, presentamos las conclusiones del trabajo de tesis y proponemos trabajos futuros.



## Capítulo 2

# Métodos de Volúmenes Finitos (MVF )

Moukalled [10] señala que los MVF transforman las ecuaciones diferenciales parciales sobre celdas o volúmenes finitos en ecuaciones algebraicas. Los métodos se inician discretizando el dominio geométrico en elementos que no se traslapen llamados volúmenes de control (volúmenes finitos). Luego, se integra la ecuación diferencial parcial sobre cada volumen y a continuación, se discretiza la ecuación integral resultante, de modo que obtenemos un sistema de ecuaciones algebraicas.

En los dos capítulos siguientes estudiamos en detalle las partes que lo componen tomando como base los aportes de Wang [22].

### 2.1. Forma integral de la ecuación de advección unidimensional.

Una ecuación diferencial parcial que modela el transporte de una sustancia bajo la influencia de la advección de un flujo, es usualmente escrita como

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \nabla \cdot (\alpha \nabla u) \quad (2.1)$$

donde  $u = u(\mathbf{x}, t) \in C^2(\Omega) \times C(0, \infty)$  es la densidad o concentración de la sustancia a ser transportada,  $\mathbf{v}$  es el campo de velocidad (supuestamente conocido),  $\alpha$  es el

coeficiente de difusión y  $\Omega \subset \mathbb{R}^d$  es el dominio computacional.

En esta tesis, supondremos que no existe la componente difusiva ( $\alpha = 0$ ), trabajaremos el caso unidimensional ( $\Omega = [0, 1]$ ), el campo de velocidades constante ( $\mathbf{v} = a$ ) y condiciones de frontera de Dirichlet y periódicas. Luego, la ecuación (2.1) se reduce a la ecuación de advección unidimensional siguiente:

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad (2.2)$$

donde  $f(u(t, x)) = f(u) = a u$  es llamado flujo y  $a$  es la velocidad de propagación que asumiremos positiva.

Tomamos una partición de tamaño  $N$  del dominio computacional  $\Omega$ :

$$\Omega = \cup_{i=1}^N I_i,$$

con  $I_i \equiv [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}]$  el volumen de control  $i$ -ésimo de esta partición. Denotamos a  $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = |I_i|$  la longitud de la celda  $I_i$ , y  $x_i = \frac{x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}}{2}$  el centroide del intervalo  $I_i$  (Ver Figura 2.1).

Para una discretización uniforme del dominio espacial con  $\Delta x_i = h$ , la forma integral de la ecuación (2.2) sobre el volumen de control  $I_i$  es

$$\int_{I_i} \frac{\partial u(t, x)}{\partial t} dx + \int_{I_i} a \frac{\partial u(t, x)}{\partial x} dx = 0. \quad (2.3)$$

Dado que los  $I_i$  son compactos y fijos en el tiempo

$$\frac{d}{dt} \int_{I_i} u(t, x) dx + a u(t, x) \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx = 0. \quad (2.4)$$

## 2.2. Discretización espacial de la forma integral de la ecuación de advección 1D.

Consideremos

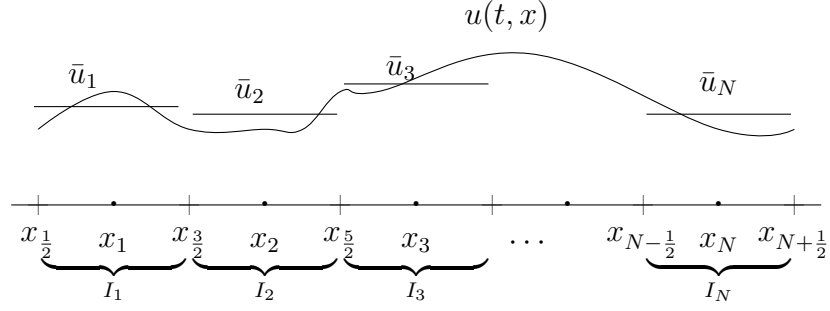


Figura 2.1: Valores medios  $\bar{u}_i$  de la función  $u(t, x)$  en cada intervalo  $I_i$  para un tiempo fijo  $t$ .

$$\bar{u}_i = \bar{u}_i(t) = \frac{1}{\Delta x_i} \int_{I_i} u(t, x) dx,$$

al valor promedio de la solución  $u(t, x)$  sobre cada intervalo  $I_i$  para un  $t$  fijo, tal como se muestra en la Figura 2.1.

Luego, la ecuación (2.4) es reescrita como

$$\begin{aligned} \frac{d\bar{u}_i}{dt} &= - \frac{a u \left( x_{i+\frac{1}{2}} \right) - a u \left( x_{i-\frac{1}{2}} \right)}{h} \\ &= - \frac{1}{h} \left( f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right) \end{aligned} \quad (2.5)$$

donde  $f_{i\pm\frac{1}{2}} = f \left( u \left( t, x_{i\pm\frac{1}{2}} \right) \right) = a u \left( t, x_{i\pm\frac{1}{2}} \right)$  son los flujos exactos en cada control  $I_i$ .

Los MVF aproximan la ecuación (2.5) introduciendo el flujo numérico

$$\hat{f}_{i\pm\frac{1}{2}} = \hat{f}_{i\pm\frac{1}{2}} \left( \bar{u}_1; \bar{u}_2, \dots, \bar{u}_N, x_{i\pm\frac{1}{2}}; t \right),$$

que depende de todos los valores promedios, de modo que para un tiempo  $t$  fijo se tiene

$$\frac{d\bar{u}_i}{dt} = - \frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right),$$

o sea

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} \left( \bar{u}_1; \bar{u}_2, \dots, \bar{u}_N, x_{i+\frac{1}{2}}; t \right) - \hat{f}_{i-\frac{1}{2}} \left( \bar{u}_1; \bar{u}_2, \dots, \bar{u}_N, x_{i-\frac{1}{2}}; t \right) \right). \quad (2.6)$$

La definición explícita del flujo numérico se basa en dos elementos fundamentales: uno que nos proporcione una reconstrucción espacial en base a los valores promedios y, a continuación, otro que nos permita calcular el flujo numérico  $\hat{f}_{i\pm\frac{1}{2}}$  en base a la reconstrucción. Luego de este proceso procedemos a realizar la iteración temporal como se establece a continuación.

### 2.3. Discretización temporal

La forma semi discreta de la ecuación (2.5); es decir, la ecuación (2.6) puede ser escrita de la siguiente manera:

$$\frac{d\bar{u}_i}{dt} = G(\bar{u}, t) = -\frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right), \quad (2.7)$$

donde  $\bar{u}$  representa a  $\bar{u}_1; \bar{u}_2, \dots, \bar{u}_N$ . No hemos colocado  $x_{i\pm\frac{1}{2}}$  en el argumento por facilidad de lectura. De este modo, la discretización espacial nos proporciona una ecuación diferencial ordinaria que describe la evolución temporal de los valores promedio. Esta evolución temporal es discretizada en esta tesis mediante esquemas de Runge-Kutta.

Jameson [7] muestra cómo los esquemas de Runge-Kutta son métodos muy efectivos para resolver ecuaciones de Euler en dominios geométricos arbitrarios, en particular para resolver una ecuación de advección unidimensional.

En estos esquemas se debe usar la condición inicial,  $u(x, 0) = g(x)$ , para evaluar los valores promedios  $\bar{u}_i^0$  en cada uno de los intervalos  $I_i$ . Un paso en la discretización temporal de tamaño  $\Delta t$ , debe proporcionarnos los nuevos valores promedio  $\bar{u}_i^1$  tal como se observa en la Figura 2.2. De manera iterativa se van calculando los  $\bar{u}_i^n$ ; es decir, los valores promedios para el tiempo  $t_n$ .

A continuación, se describen algunos esquema de Runge-Kutta que se diferencian en la manera en cómo se calcula  $\bar{u}_i^{n+1}$  a partir de  $\bar{u}_i^n$  y por el orden de su aproximación.

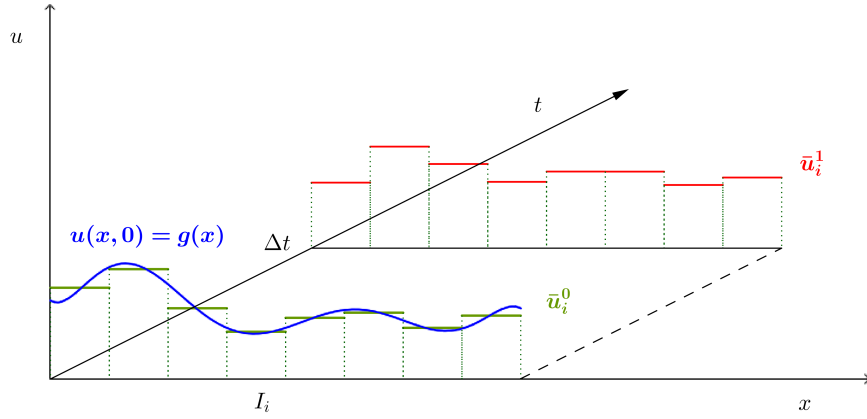


Figura 2.2: Evolución en el tiempo de los valores medio  $\bar{u}_i$

### Esquema de Runge-Kutta de cuarto orden con cuatro etapas

Este esquema de integración de cuarto orden en el tiempo, popularizado por Jameson [7], por tener una amplia región de estabilidad, tiene una condición CFL (Courant–Friedrichs–Lewy) para la ecuación de advección unidimensional

$$CFL = \left| \frac{a\Delta t}{\Delta x} \right| \leq 2\sqrt{2}.$$

Una iteración temporal para conseguir un paso del tiempo de  $\bar{u}_i^n$  a  $\bar{u}_i^{n+1}$  en el intervalo  $i$  es escrito como sigue:

$$\begin{aligned} \bar{u}_i^{(1)} &= \bar{u}_i^n - \frac{\Delta t}{2} G(\bar{u}^n, t), \\ \bar{u}_i^{(2)} &= \bar{u}_i^n - \frac{\Delta t}{2} G\left(\bar{u}^{(1)}, t + \frac{\Delta t}{2}\right), \\ \bar{u}_i^{(3)} &= \bar{u}_i^n - \Delta t G\left(\bar{u}^{(2)}, t + \frac{\Delta t}{2}\right), \\ \bar{u}_i^{n+1} &= \bar{u}_i^n - \frac{\Delta t}{6} [G(\bar{u}^n) + 2G(\bar{u}^{(1)}) + 2G(\bar{u}^{(2)}) + G(\bar{u}^{(3)})], \end{aligned}$$

para cada  $i = 1, \dots, N$ ; donde  $\bar{u}_i^{(1)}, \bar{u}_i^{(2)}, \bar{u}_i^{(3)}$  y  $\bar{u}_i^{(n+1)}$  son las cuatro etapas intermedias y  $G$  es definido en la ecuación (2.7).

## Esquema de Runge-Kutta SSPRK (Strong-Stability-Preserving Runge-Kutta )

Como indica Fehlberg en [4] la estabilidad fuerte<sup>1</sup> de los esquema de Runge-Kutta SSPRK, se preservan para el Método de Euler hacia adelante con cualquier norma o semi norma. Los coeficientes que caracterizan a estos esquemas no son únicos y Shu [16] muestra versiones que maximizan el CFL para esquemas de segundo y tercer orden. Estos son presentados a continuación.

**SSPRK(2,2)** : Dos etapas de segundo orden donde

$$\begin{aligned}\bar{u}_i^{(1)} &= \bar{u}_i^n - \Delta t G(\bar{u}^n, t) \\ \bar{u}_i^{n+1} &= \frac{1}{2}\bar{u}_i^n + \frac{1}{2}\bar{u}_i^{(1)} - \frac{1}{2}\Delta t G(\bar{u}^{(n)}, t + \Delta t)\end{aligned}$$

para cada  $i = 1, \dots, N$  ;  $n = 1, 2, \dots$  y con  $G$  definido por la ecuación (2.7).

**SSPRK(3,3)** : Tres etapas de tercer orden donde

$$\begin{aligned}\bar{u}_i^{(1)} &= \bar{u}_i^n - \Delta t G(\bar{u}^n, t) \\ \bar{u}_i^{(2)} &= \frac{3}{4}\bar{u}_i^n + \frac{1}{4}\bar{u}_i^{(1)} - \frac{1}{4}\Delta t G(\bar{u}^{(1)}, t + \Delta t) \\ \bar{u}_i^{n+1} &= \frac{1}{3}\bar{u}_i^n + \frac{2}{3}\bar{u}_i^{(2)} - \frac{2}{3}\Delta t G\left(\bar{u}^{(2)}, t + \frac{\Delta t}{2}\right)\end{aligned}$$

para cada  $i = 1, \dots, N$ ;  $n = 1, 2, \dots$  y con  $G$  definido por la ecuación (2.7). Gottlie [6] denomina a este esquema Runge-Kutta TVD (Total Variation Diminishing).

En la implementación realizada usaremos únicamente el esquema de tercer orden.

*Observación 1.* Debemos tener en cuenta que cualquiera sea el orden de la reconstrucción, el orden en la solución dependerá también del orden de la discretización temporal; es decir, si consideramos el error de aproximación, éste debe ser tal que

$$\|u - \tilde{u}\| = \mathcal{O}(\Delta x^k) + \mathcal{O}(\Delta t^q),$$

<sup>1</sup>Ruuth [15] define una secuencia  $\{U^n\}$  con estabilidad fuerte cuando dada cualquier semi-norma  $\|\cdot\|$ , se tiene  $\|U^{n+1}\| \leq \|U^n\|$  para todo  $n \geq 0$ .

donde  $k$  es el orden de la discretización espacial y  $q$ , el orden de la discretización temporal. Así, el orden de aproximación será  $\min\{k, q\}$ .

En los dos capítulos siguientes concentramos nuestra atención en el diseño de dos componentes esenciales en los MVF: la reconstrucción y el cálculo del flujo numérico.



## Capítulo 3

### MVF Compacto: Reconstrucción

Una de las características de un método de volúmenes finitos de alto orden es el uso de reconstrucciones de alto orden. En este capítulo ponemos énfasis en el método propuesto por Wang [22], donde la reconstrucción es compacta; esto es, para el caso unidimensional, dado un intervalo  $I_i$  se usa el estencil  $S_i = \{I_{i-1}; I_{i+1}\}$  para alcanzar cualquier orden de aproximación. Otros métodos de volúmenes finitos necesitan más vecinos del intervalo  $I_i$  conforme se requiere ordenes más altos de aproximación. Para una introducción a esquemas de reconstrucción ver Olliver-Gooch [11] .

#### 3.1. Reconstrucción de mínimos cuadrados compacto

Matemáticamente el problema de reconstrucción en el esquema de VF puede ser establecido como sigue:

Dado el valor promedio

$$\bar{u}_j = \frac{1}{\Delta x_j} \int_{I_j} u(x) dx$$

de una solución  $u(x)$  sobre cada volumen de control  $I_j \subset \Omega$ , debemos construir un polinomio de grado  $k$  que aproxima a la solución  $u$  sobre el volumen de control  $I_i$ .



Desarrollamos la serie de Taylor en torno de  $x_i$  de modo que

$$u(x) = u(x_i) + \frac{du}{dx}(x_i)(x - x_i) + \frac{d^2u}{dx^2}(x_i)\frac{(x - x_i)^2}{2!} + \frac{d^3u}{dx^3}(x_i)\frac{(x - x_i)^3}{3!} + \dots \quad (3.1)$$

Tomando el valor medio sobre  $I_i$  tenemos

$$\begin{aligned} \frac{1}{\Delta x_i} \int_{I_i} u(x) dx &= \underbrace{u(x_i)}_{c_0} + \underbrace{\frac{du}{dx}(x_i)}_{c_1} \frac{1}{\Delta x_i} \int_{I_i} (x - x_i) dx + \underbrace{\frac{d^2u}{dx^2}(x_i)}_{c_2} \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^2}{2!} dx + \\ &\quad \underbrace{\frac{d^3u}{dx^3}(x_i)}_{c_3} \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^3}{3!} dx + \dots \end{aligned}$$

Por tanto

$$\begin{aligned} \bar{u}_i &= u(x_i) + c_1 \frac{1}{\Delta x_i} \int_{I_i} (x - x_i) dx + c_2 \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^2}{2!} dx + \\ &\quad c_3 \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^3}{3!} dx + \dots \end{aligned}$$

donde  $c_l = \frac{d^l u}{dx^l}(x_i)$ .

Ahora despejando  $u(x_i)$  de la expresión anterior y reemplazando en (3.1) obtenemos

$$\begin{aligned} u(x) &= \bar{u}_i + c_1 \left\{ (x - x_i) - \frac{1}{\Delta x_i} \int_{I_i} (x - x_i) dx \right\} + c_2 \left\{ \frac{(x - x_i)^2}{2!} - \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^2}{2!} dx \right\} + \\ &\quad c_3 \left\{ \frac{(x - x_i)^3}{3!} - \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)^3}{3!} dx \right\} + \dots \end{aligned}$$

Reordenando términos obtenemos

$$\begin{aligned} u(x) &= \bar{u}_i + c_1 \Delta x_i \left\{ \frac{(x - x_i)}{\Delta x_i} - \frac{1}{\Delta x_i} \int_{I_i} \frac{(x - x_i)}{\Delta x_i} dx \right\} \\ &\quad + \frac{c_2 (\Delta x_i)^2}{2!} \left\{ \left( \frac{x - x_i}{\Delta x_i} \right)^2 - \frac{1}{\Delta x_i} \int_{I_i} \left( \frac{x - x_i}{\Delta x_i} \right)^2 dx \right\} \\ &\quad + \frac{c_3 (\Delta x_i)^2}{3!} \left\{ \left( \frac{x - x_i}{\Delta x_i} \right)^3 - \frac{1}{\Delta x_i} \int_{I_i} \left( \frac{x - x_i}{\Delta x_i} \right)^3 dx \right\} + \dots, \end{aligned}$$

donde  $c_l = \frac{d^l u}{dx^l}(x_i)$ .

Luego, tenemos una aproximación polinomial de grado  $k$  para  $u(x)$  en torno del  $x_i$  del intervalo  $I_i$ ,

$$u(x) \approx u_i(x) = \bar{u}_i + \sum_{l=1}^k u_i^l \varphi_{l,i}(x) \quad (3.2)$$

donde

$$u_i^l = \frac{1}{l!} \frac{d^l u}{dx^l}(x_i) (\Delta x_i)^l \quad (3.3)$$

es el coeficiente de reconstrucción a ser determinado y  $\varphi_{l,i}(x)$  es la base de media cero definida por

$$\varphi_{l,i}(x) = (\delta x_i)^l - \frac{1}{\Delta x_i} \int_{I_i} (\delta x_i)^l dx, \quad \text{donde } \delta x = \frac{x - x_i}{\Delta x_i}. \quad (3.4)$$

o equivalentemente

$$\varphi_{l,i}(x) = \begin{cases} \left( \frac{x-x_i}{\Delta x_i} \right)^l - \frac{1}{(l+1)2^l}, & l \bmod 2 = 0 \\ \left( \frac{x-x_i}{\Delta x_i} \right)^l, & \text{caso contrario.} \end{cases}$$

*Observación 2.* La base presentada arriba es normalizada por la longitud de la celda  $\Delta x_i$  para evitar el crecimiento del número de condicionamiento de la matriz de reconstrucción al realizar un refinamiento de la malla [5]. Al tener bases de media cero, la reconstrucción polinomial  $u_i(x)$  es siempre conservativa, en el sentido que

$$\frac{1}{\Delta x_i} \int_{I_i} u_i(x) dx = \bar{u}_i.$$

Esto quiere decir que los valores promedios del polinomio de reconstrucción coinciden con el valor promedio de la solución.

*Observación 3.* Existen  $k$  coeficientes  $u_i^l$ ,  $l = 1, 2, \dots, k$  en (3.3) para ser determinados en el procedimiento de reconstrucción, donde  $k$  definirá el orden de reconstrucción.

En los algoritmos de reconstrucción conocidos como k-exactos, ENO y WENO

para esquemas de VF, los coeficientes  $u_i^l$ ,  $l = 1, 2, \dots, k$  son determinados por conservación de los valores promedios correspondientes a los intervalos solución  $I_i$ , cuando los intervalos vecinos de  $I_i$  pertenecientes al estencil de reconstrucción con una solución suficientemente suave. Más específicamente, sobre el estencil  $S_i$ , el cual consiste de  $n_i \geq k$  celdas vecinas de  $I_i$  y  $I_i \notin S_i$ , se requiere que

$$\frac{1}{\Delta x_j} \int_{I_j} u_i(x) dx = \bar{u}_i + \sum_{l=1}^k u_i^l \left( \frac{1}{\Delta x_j} \int_{I_j} \varphi_{l,i}(x) dx \right) = \bar{u}_j \quad (3.5)$$

para todo  $I_j \in S_i$ . Este procedimiento resulta en un sistema de  $n_i$  ecuaciones lineales. Si  $n_i > k$ , el sistema de ecuaciones es uno sobre determinado y necesita ser resuelto usando el método de mínimos cuadrados. El principal problema de esta aproximación es que el número de celdas en el estencil  $S_i$  se incrementa con el orden  $k$ . Esto produce un gran estencil para esquema de VF de órdenes altos. Como afirma Wang[22], grandes estenciles en esquemas de VF de órdenes altos destruyen la compacidad del esquema y reducen la eficiencia computacional especialmente en la programación en paralelo, lo cual ha sido el cuello de botella para métodos de VF de órdenes altos sobre mallas no estructuradas.

La idea básica de la reconstrucción propuesta es la siguiente: Junto con las relaciones que describe la ecuación (3.5), las relaciones adicionales son reemplazadas por el requerimiento de conservación sobre  $S_i$  de los valores promedio de las derivadas de varios órdenes  $m$  del polinomio de reconstrucción. Esto quiere decir que para todos los  $I_j \in S_i$ ,

$$\frac{1}{\Delta x_j} \int_{I_j} \frac{d^m u_i(x)}{dx^m} dx = \frac{1}{\Delta x_j} \int_{I_j} \frac{d^m u_j(x)}{dx^m} dx, \quad 0 \leq m \leq M \quad (3.6)$$

donde  $M \leq k$  corresponde a la derivada de mayor orden. Reemplazando la ecuación (3.2) en (3.6) obtenemos

$$\sum_{l=1}^k u_i^l \left( \frac{1}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,i}(x)}{dx^m} dx \right) =$$

$$\delta_m^0 (\bar{u}_j - \bar{u}_i) + \sum_{l=1}^k u_j^l \left( \frac{1}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,j}(x)}{dx^m} \right) dx \quad (3.7)$$

donde  $0 \leq m \leq M$ . Definimos  $\mathbf{u}_i = [u_i^1, u_i^2, \dots, u_i^k]^T$  y reescribimos la ecuación (3.7) en forma matricial

$$\mathbf{A}_j^i \mathbf{u}_i = \mathbf{B}_j^i \mathbf{u}_j + \mathbf{b}_j^i \quad (3.8)$$

donde

$$(\mathbf{A}_j^i)_{ml} = \frac{1}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,i}(x)}{dx^m} dx, \quad m = 0, \dots, M; \quad l = 1, \dots, k; \quad (3.9)$$

$$(\mathbf{B}_j^i)_{ml} = \frac{1}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,j}(x)}{dx^m} dx, \quad m = 0, \dots, M; \quad l = 1, \dots, k; \quad (3.10)$$

$$(\mathbf{b}_j^i)_m = \delta_m^0 (\bar{u}_j - \bar{u}_i), \quad m = 0, \dots, M.$$

Definimos  $S_i$  como el estencil compacto que únicamente toma los vecino directos o las caras vecinas de  $I_i$ . Para el caso unidimensional  $S_i = \{I_{i-1}, I_{i+1}\}$ , y la ecuación (3.8) se reduce a

$$\mathbf{A}_{i-1}^i \mathbf{u}_i = \mathbf{B}_{i-1}^i \mathbf{u}_{i-1} + \mathbf{b}_{i-1}^i \quad (3.11)$$

y

$$\mathbf{A}_{i+1}^i \mathbf{u}_i = \mathbf{B}_{i+1}^i \mathbf{u}_{i+1} + \mathbf{b}_{i+1}^i. \quad (3.12)$$

Puesto que  $\mathbf{u}_{i-1}, \mathbf{u}_i$  y  $\mathbf{u}_{i+1}$  son todas variables, las ecuaciones (3.11)–(3.12) deben resolverse implícitamente formando un sistema de ecuaciones lineales por bloques. En el presente trabajo elegimos  $M = k - 1$ , valor sugerido por Wang [22]. Como el sistema (3.11)–(3.12) es sobre determinado no puede resolverse directamente. Luego, resolvemos el sistema por mínimos cuadrados. Las ecuaciones (3.11)–(3.12) pueden reordenarse en

$$\mathbf{A}_i \mathbf{u}_i = \mathbf{B}_{i-1} \mathbf{u}_{i-1} + \mathbf{B}_{i+1} \mathbf{u}_{i+1} + \mathbf{b}_i \quad (3.13)$$

donde

$$\mathbf{A}_i = \begin{pmatrix} \mathbf{A}_{i-1}^i \\ \mathbf{A}_{i+1}^i \end{pmatrix}, \quad \mathbf{B}_{i-1} = \begin{pmatrix} \mathbf{B}_{i-1}^i \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{B}_{i+1} = \begin{pmatrix} \mathbf{0} \\ \mathbf{B}_{i+1}^i \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} \mathbf{b}_{i-1}^i \\ \mathbf{b}_{i+1}^i \end{pmatrix} \quad (3.14)$$

El siguiente sistema es resuelto usando el método de mínimos cuadrados

$$(\mathbf{A}_i^T \mathbf{A}_i) \mathbf{u}_i = (\mathbf{A}_i^T \mathbf{B}_{i-1}) \mathbf{u}_{i-1} + (\mathbf{A}_i^T \mathbf{B}_{i+1}) \mathbf{u}_{i+1} + \mathbf{A}_i^T \mathbf{b}_i. \quad (3.15)$$

Para la reconstrucción se puede incorporar funciones peso  $w_{i,m}$ , con  $m = 0, \dots, M$  de la forma

$$w_{i,m} = \begin{cases} 1, & m = 0 \\ w_m (\Delta x_i)^m, & m > 0 \end{cases}$$

en las ecuaciones (3.9) y (3.10). El propósito es mejorar las propiedades espectrales. Así, redefinimos

$$(\mathbf{A}_j^i)_{ml} = \frac{w_{i,m}}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,i}(x)}{dx^m} dx, \quad m = 0, \dots, M; \quad l = 1, \dots, k \quad (3.16)$$

$$(\mathbf{B}_j^i)_{ml} = \frac{w_{i,m}}{\Delta x_j} \int_{I_j} \frac{d^m \varphi_{l,j}(x)}{dx^m} dx, \quad m = 0, \dots, M; \quad l = 1, \dots, k. \quad (3.17)$$

Entonces, si definimos

$$\mathbf{D}_i = \mathbf{A}_i^T \mathbf{A}_i \quad \mathbf{E}_i = -\mathbf{A}_i^T \mathbf{B}_{i-1}, \quad \mathbf{F}_i = -\mathbf{A}_i^T \mathbf{B}_{i+1}, \quad \mathbf{g}_i = \mathbf{A}_i^T \mathbf{b}_i$$

la ecuación (3.15) es equivalente a

$$\mathbf{E}_i \mathbf{u}_{i-1} + \mathbf{D}_i \mathbf{u}_i + \mathbf{F}_i \mathbf{u}_{i+1} = \mathbf{g}_i \quad i = 2, \dots, N-1 \quad (3.18)$$

obteniéndose un sistema tridiagonal en bloques de la forma

$$\begin{pmatrix} \mathbf{D}_2 & \mathbf{F}_2 & & & \\ \mathbf{E}_3 & \mathbf{D}_3 & \mathbf{F}_3 & & \\ & \mathbf{E}_4 & & \ddots & \\ & & \ddots & \ddots & \mathbf{F}_{N-2} \\ & & & \mathbf{E}_{N-1} & \mathbf{D}_{N-1} \end{pmatrix} \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \vdots \\ \mathbf{u}_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_2^* \\ \mathbf{g}_3 \\ \mathbf{g}_3 \\ \vdots \\ \mathbf{g}_{N-1}^* \end{pmatrix} \quad (3.19)$$

el mismo que puede ser resuelto con un algoritmo equivalente al sistema tridiagonal por descomposición  $LU$ . La matriz de coeficientes es la que llamamos matriz de reconstrucción,

### 3.1.1. Condiciones de frontera de Dirichlet

Para esta condición de frontera tenemos

$$u(0, t) = u_0(t), \quad t \in [0, 1].$$

Ahora en la ecuación (3.19) se tienen resueltos los coeficientes  $\mathbf{u}_i$  de los intervalos interiores, pero aún falta determinar los coeficientes  $\mathbf{u}_1$  para la reconstrucción del primer intervalo  $I_1$  y  $\mathbf{u}_N$  para la reconstrucción del intervalo  $I_N$ . En este caso, para una condición de frontera de Dirichlet se considera un estencil vecino a la derecha de la primera celda y en la última únicamente tenemos un estencil vecino izquierdo; es decir,  $S_1 = \{I_2\}$  y  $S_N = \{I_{N-1}\}$ . Entonces, para la primera celda la ecuación (3.11) no sería necesaria (Para  $i = 1$ ,  $\mathbf{A}_{i-1}^1 = \mathbf{0} = \mathbf{B}_{i-1}^1$ ,  $\mathbf{b}_0^1 = \mathbf{0}$ ) y sólo la ecuación (3.12) debería ser tomada en cuenta. Así en la ecuación (3.18) el término

$$\mathbf{E}_1 = -\mathbf{A}_1^T \mathbf{B}_0 = -\begin{pmatrix} \mathbf{0} \\ \mathbf{A}_2^1 \end{pmatrix}^T \begin{pmatrix} \mathbf{B}_0^1 \\ \mathbf{0} \end{pmatrix} = \mathbf{0};$$

del mismo modo,  $\mathbf{F}_N = \mathbf{0}$  y

$$\mathbf{g}_1 = \mathbf{A}_1^T \mathbf{b}_1 = \begin{pmatrix} \mathbf{0} \\ \mathbf{A}_2^1 \end{pmatrix}^T \begin{pmatrix} \mathbf{0} \\ \mathbf{b}_2^1 \end{pmatrix} = \mathbf{A}_2^{1T} \mathbf{b}_2^1, \quad (3.20)$$

y

$$\mathbf{g}_N = \mathbf{A}_N^T \mathbf{b}_N = \begin{pmatrix} \mathbf{A}_{N-1}^N \\ \mathbf{0} \end{pmatrix}^T \begin{pmatrix} \mathbf{b}_{N-1}^N \\ \mathbf{0} \end{pmatrix} = (\mathbf{A}_{N-1}^N)^T \mathbf{b}_{N-1}^N. \quad (3.21)$$

Por tanto, con estas consideraciones, en el sistema (3.19) se pueden agregar las dos ecuaciones

$$\mathbf{D}_1 \mathbf{u}_1 + \mathbf{F}_1 \mathbf{u}_2 = \mathbf{g}_1 \quad \text{y} \quad \mathbf{E}_N \mathbf{u}_{N-1} + \mathbf{D}_N \mathbf{u}_N = \mathbf{g}_N, \quad (3.22)$$

de modo que el sistema quedaría expresado en su forma matricial como sigue:

$$\begin{pmatrix} \mathbf{D}_1 & \mathbf{F}_1 & & & & \\ \mathbf{E}_2 & \mathbf{D}_2 & \mathbf{F}_2 & & & \\ & \mathbf{E}_3 & \mathbf{D}_3 & \mathbf{F}_3 & & \\ & & \mathbf{E}_4 & & \ddots & \\ & & & \ddots & \ddots & \mathbf{F}_{N-2} \\ & & & & \mathbf{E}_{N-1} & \mathbf{D}_{N-1} & \mathbf{F}_{N-1} \\ & & & & & \mathbf{E}_N & \mathbf{D}_N \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \\ \mathbf{u}_{N-1} \\ \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \vdots \\ \mathbf{g}_{N-1} \\ \mathbf{g}_N \end{pmatrix}$$

donde  $\mathbf{g}_1$  y  $\mathbf{g}_N$  están dadas por las ecuaciones (3.20) y (3.21).

### 3.1.2. Condiciones de frontera periódicas

Para este tipo de condiciones se cumple que

$$u(0, t) = u(1, t), \quad t \in [0, 1]$$

Del mismo modo que en la sección anterior, en la ecuación (3.18) aún falta determinar los coeficientes  $\mathbf{u}_1$  para la reconstrucción del primer intervalo  $I_1$  y  $\mathbf{u}_N$  para la reconstrucción del intervalo  $I_N$ . El tratamiento para la primera celda y la última será diferenciado e introduciremos intervalos fantasmas para su solución (Ver Figura 3.1).

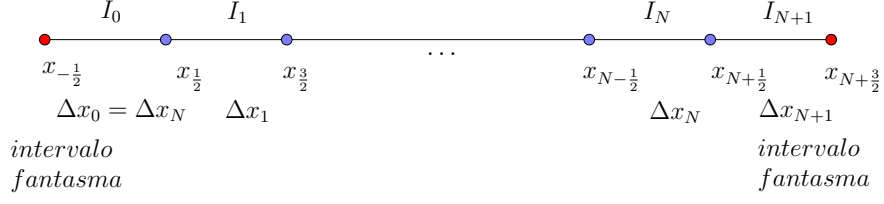


Figura 3.1: Tratamiento de las condiciones de frontera periódicas

Para el primer intervalo tenemos que  $i = 1$ ,  $\Delta x_0 = x_{N-\frac{1}{2}} - x_{N+\frac{1}{2}}$  de igual longitud que el último intervalo  $I_N$ , de allí que,  $I_0 = [x_{-\frac{1}{2}}; x_{\frac{1}{2}}]$ ,  $x_{-\frac{1}{2}} = x_{\frac{1}{2}} - \Delta x_0$ . Luego, de las ecuaciones (3.16) y (3.17)

$$(\mathbf{A}_0^1)_{ml} = \frac{w_{1,m}}{\Delta x_0} \int_{I_0} \frac{d^m \varphi_{l,1}(x)}{\partial x^m} dx = \frac{w_{1,m}}{\Delta x_0} \int_{x_{-\frac{1}{2}}}^{x_{\frac{1}{2}}} \frac{d^m \varphi_{l,1}(x)}{dx^m} dx,$$

$$(\mathbf{B}_0^1)_{ml} = \frac{w_{i,m}}{\Delta x_0} \int_{I_0} \frac{d^m \varphi_{l,0}(x)}{\partial x^m} dx = \frac{w_{i,m}}{\Delta x_0} \int_{x_{-\frac{1}{2}}}^{x_{\frac{1}{2}}} \frac{d^m \varphi_{l,0}(x)}{dx^m} dx,$$

para  $m = 0, \dots, M$ ;  $l = 1, \dots, k$

$$(\mathbf{b}_0^1)_m = \delta_m^0 (\bar{u}_0 - \bar{u}_1), \quad m = 0, \dots, M$$

donde  $\bar{u}_0 = \bar{u}_N$ .

De manera similar para el último intervalo con  $i = N$ ,  $\Delta x_{N+1} = x_{\frac{3}{2}} - x_{\frac{1}{2}}$  de igual longitud que el primer intervalo. De allí que  $I_{N+1} = [x_{N+\frac{3}{2}}; x_{N+\frac{1}{2}}]$ ,  $x_{N+\frac{3}{2}} = x_{N+\frac{1}{2}} + \Delta x_{N+1}$ . Luego, de las ecuaciones (3.16) y (3.17) son expresadas como

$$(\mathbf{A}_{N+1}^N)_{ml} = \frac{w_{1,m}}{\Delta x_{N+1}} \int_{I_{N+1}} \frac{d^m \varphi_{l,N}(x)}{\partial x^m} dx = \frac{w_{1,m}}{\Delta x_{N+1}} \int_{x_{N+\frac{1}{2}}}^{x_{N+\frac{3}{2}}} \frac{d^m \varphi_{l,N}(x)}{\partial x^m} dx,$$

$$(\mathbf{B}_{N+1}^N)_{ml} = \frac{w_{i,m}}{\Delta x_{N+1}} \int_{I_{N+1}} \frac{d^m \varphi_{l,N+1}(x)}{\partial x^m} dx = \frac{w_{i,m}}{\Delta x_{N+1}} \int_{x_{N+\frac{1}{2}}}^{x_{N+\frac{3}{2}}} \frac{d^m \varphi_{l,N+1}(x)}{\partial x^m} dx,$$

para  $m = 0, \dots, M$ ;  $l = 1, \dots, k$  y

$$(\mathbf{b}_{N+1}^N)_m = \delta_m^0 (\bar{u}_{N+1} - \bar{u}_N), \quad m = 0, \dots, M$$



donde  $\bar{u}_{N+1} = \bar{u}_1$ .

Así, tenemos un sistema matricial similar al de la ecuación (3.18), la diferencia es que hemos definido para  $i = 1$  e  $i = N$  la siguiente relación:

$$\mathbf{E}_i \mathbf{u}_{i-1} + \mathbf{D}_i \mathbf{u}_i + \mathbf{F}_i \mathbf{u}_{i+1} = \mathbf{g}_i \quad i = 1, \dots, N, \quad (3.23)$$

donde  $\mathbf{E}_i$ ,  $\mathbf{D}_i$ ,  $\mathbf{F}_i$  y  $\mathbf{g}_i$  son definidos en la subsección anterior. De este modo introducimos las matrices  $\mathbf{E}_1$  y  $\mathbf{F}_N$  en la matriz de reconstrucción obteniendo el sistema tridiagonal cíclico

$$\begin{pmatrix} \mathbf{D}_1 & \mathbf{F}_1 & & & & \mathbf{E}_1 \\ \mathbf{E}_2 & \mathbf{D}_2 & \mathbf{F}_2 & & & \\ & \mathbf{E}_3 & \mathbf{D}_3 & \mathbf{F}_3 & & \\ & & \mathbf{E}_4 & & \ddots & \\ & & & \ddots & \ddots & \mathbf{F}_{N-2} \\ & & & & \mathbf{E}_{N-1} & \mathbf{D}_{N-1} & \mathbf{F}_{N-1} \\ \mathbf{F}_N & & & & & \mathbf{E}_N & \mathbf{D}_N \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \\ \mathbf{u}_{N-1} \\ \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \vdots \\ \mathbf{g}_{N-1} \\ \mathbf{g}_N \end{pmatrix}$$

donde  $\mathbf{u}_0 = \mathbf{u}_N$  y  $\mathbf{u}_{N+1} = \mathbf{u}_1$ .

### 3.1.3. Consideraciones de la reconstrucción

**Primera** La novedad del método presentado por Wang [22] son las relaciones descritas por las ecuaciones (3.6) y (3.7) donde se puede establecer un orden arbitrario usando únicamente celdas vecinas para su formulación. Es por ello que este procedimiento es considerado compacto y junto con la ecuación (3.18), derivado del método de mínimos cuadrados para sistemas sobre determinados, la reconstrucción es llamada reconstrucción por mínimos cuadrados compacto (CLS de las siglas en inglés de Compact Least-Square). El método de volúmenes finitos que usa CLS para su reconstrucción es referenciado como método CLSFV.

**Segunda** En el trabajo de Wang [22], se elige  $M = k - 1$ , de modo que el número de ecuaciones en la ecuación (3.13) dobla al número de variables. Los

parámetros libres  $w_m$  que se toman en cuenta en (3.16) y (3.17) Wang los usa para optimizar las propiedades espectrales.

### 3.2. Matrices de reconstrucción explícitas.

En esta sección mostramos expresiones explícitas de los bloques que componen la matriz de reconstrucción, obtenidas en la sección anterior. Estas matrices bloque están descritas en las ecuaciones (3.16) y (3.17); y las obtendremos con la librería de procesamiento simbólico SymPy de Python (Ver Anexo). Obtendremos aquellas de segundo orden ( $k = 1$ ) hasta de séptimo orden ( $k = 6$ ) y, por simplicidad, sobre mallas uniformes ( $\Delta x_i = \Delta x$ ).

#### Reconstrucción lineal ( $k = 1$ )

$$\mathbf{A}_i = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{B}_i = \mathbf{0}, \quad \mathbf{B}_{i+1} = \mathbf{0},$$

$$\mathbf{u}_i = u_i^1, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ \bar{u}_{i+1} - \bar{u}_i \end{bmatrix} \quad (3.24)$$

#### Reconstrucción cuadrática ( $k = 2$ )

$$\mathbf{A}_i = \begin{bmatrix} -1 & 1 \\ w_1 & -2w_1 \\ 1 & 1 \\ w_1 & 2w_1 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 0 & 0 \\ w_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{i+1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ w_1 & 0 \end{bmatrix}$$

$$\mathbf{u}_i = \begin{bmatrix} u_i^1 \\ u_i^2 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ 0 \\ \bar{u}_{i+1} - \bar{u}_i \\ 0 \end{bmatrix} \quad (3.25)$$

#### Reconstrucción cúbica ( $k = 3$ )

$$\begin{aligned}
\mathbf{A}_i &= \begin{bmatrix} -1 & 1 & -5/4 \\ w_1 & -2w_1 & 13w_1/4 \\ 0 & 2w_2 & -6w_2 \\ 1 & 1 & 5/4 \\ w_1 & 2w_1 & 13w_1/4 \\ 0 & 2w_2 & 6w_2 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 0 & 0 & 0 \\ w_1 & 0 & w_1/4 \\ 0 & 2w_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{i+1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ w_1 & 0 & w_1/4 \\ 0 & 2w_2 & 0 \end{bmatrix} \\
\mathbf{u}_i &= \begin{bmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ 0 \\ 0 \\ \bar{u}_{i+1} - \bar{u}_i \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{3.26}$$

**Reconstrucción cuarta** ( $k = 4$ )

$$\begin{aligned}
\mathbf{A}_i &= \begin{bmatrix} -1 & 1 & -5/4 & 3/2 \\ w_1 & -2w_1 & 13w_1/4 & -5w_1 \\ 0 & 2w_2 & -6w_2 & 13w_2 \\ 0 & 0 & 6w_3 & -24w_3 \\ 1 & 1 & 5/4 & 3/2 \\ w_1 & 2w_1 & 13w_1/4 & 5w_1 \\ 0 & 2w_2 & 6w_2 & 13w_2 \\ 0 & 0 & 6w_3 & 24w_3 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ w_1 & 0 & w_1/4 & 0 \\ 0 & 2w_2 & 0 & w_2 \\ 0 & 0 & 6w_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{B}_{i+1} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ w_1 & 0 & w_1/4 & 0 \\ 0 & 2w_2 & 0 & w_2 \\ 0 & 0 & 6w_3 & 0 \end{bmatrix}, \quad \mathbf{u}_i = \begin{bmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \\ u_i^4 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \\ \bar{u}_{i+1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{3.27}$$

**Reconstrucción quinta** ( $k = 5$ )

$$\begin{aligned}
\mathbf{A}_i &= \begin{bmatrix} -1 & 1 & -5/4 & 3/2 & -91/48 \\ w_1 & -2w_1 & 13w_1/4 & -5w_1 & 121w_1/16 \\ 0 & 2w_2 & -6w_2 & 13w_2 & -25w_2 \\ 0 & 0 & 6w_3 & -24w_3 & 64w_3 \\ 0 & 0 & 0 & 24w_4 & -120w_4 \\ 1 & 1 & 5/4 & 3/2 & -91/48 \\ w_1 & 2w_1 & 13w_1/4 & 5w_1 & 121w_1/16 \\ 0 & 2w_2 & 6w_2 & 13w_2 & -25w_2 \\ 0 & 0 & 6w_3 & 24w_3 & 64w_3 \\ 0 & 0 & 0 & 24w_4 & 120w_4 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ w_1 & 0 & w_1/4 & 0 & w_1/16 \\ 0 & 2w_2 & 0 & w_2 & 0 \\ 0 & 0 & 6w_3 & 0 & 5w_3 \\ 0 & 0 & 0 & 24w_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{B}_{i+1} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ w_1 & 0 & w_1/4 & 0 & w_1/16 \\ 0 & 2w_2 & 0 & w_2 & 0 \\ 0 & 0 & 6w_3 & 0 & 5w_3 \\ 0 & 0 & 0 & 24w_4 & 0 \end{bmatrix}, \quad \mathbf{u}_i = \begin{bmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \\ u_i^4 \\ u_i^5 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \\ 0 \\ \bar{u}_{i+1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.28)
\end{aligned}$$

Reconstrucción sexta ( $k = 6$ )

$$\begin{aligned}
\mathbf{A}_i &= \begin{bmatrix} -1 & 1 & -\frac{5}{4} & \frac{3}{2} & -\frac{91}{48} & \frac{39}{16} \\ w_1 & -2w_1 & \frac{13w_1}{4} & -5w_1 & \frac{121w_1}{16} & -\frac{91w_1}{16} \\ 0 & 2w_2 & -6w_2 & 13w_2 & -25w_2 & \frac{363w_2}{8} \\ 0 & 0 & 6w_3 & -24w_3 & -150w_3 & -150w_3 \\ 0 & 0 & 0 & 24w_4 & -120w_4 & 390w_4 \\ 0 & 0 & 0 & 0 & 120w_5 & -720w_5 \\ 1 & 1 & \frac{5}{4} & \frac{3}{2} & -\frac{91}{48} & \frac{39}{16} \\ w_1 & 2w_1 & 13w_1/4 & 5w_1 & 121w_1/16 & -\frac{91w_1}{16} \\ 0 & 2w_2 & 6w_2 & 13w_2 & 25w_2 & \frac{363w_2}{8} \\ 0 & 0 & 6w_3 & 24w_3 & 64w_3 & 150w_3 \\ 0 & 0 & 0 & 24w_4 & 120w_4 & 390w_4 \\ 0 & 0 & 0 & 0 & 120w_5 & 720w_5 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ w_1 & 0 & \frac{w_1}{4} & 0 & \frac{w_1}{16} & 0 \\ 0 & 2w_2 & 0 & w_2 & 0 & \frac{3w_2}{8} \\ 0 & 0 & 6w_3 & 0 & 5w_3 & 0 \\ 0 & 0 & 0 & 24w_4 & 0 & 30w_4 \\ 0 & 0 & 0 & 0 & 120w_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{B}_{i+1} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ w_1 & 0 & \frac{w_1}{4} & 0 & \frac{w_1}{16} & 0 \\ 0 & 2w_2 & 0 & w_2 & 0 & \frac{3w_2}{8} \\ 0 & 0 & 6w_3 & 0 & 5w_3 & 0 \\ 0 & 0 & 0 & 24w_4 & 0 & 30w_4 \\ 0 & 0 & 0 & 0 & 120w_5 & 0 \end{bmatrix}, \quad \mathbf{u}_i = \begin{bmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \\ u_i^4 \\ u_i^5 \\ u_i^6 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} \bar{u}_{i-1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \bar{u}_{i+1} - \bar{u}_i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.29)
\end{aligned}$$

### 3.3. Validación numérica

En esta sección, veremos un ejemplo de cómo se realiza la reconstrucción lineal y cuadrática de una función trigonométrica  $u(x) = \cos(\pi x)$  en el dominio  $[-1; 1]$ , para una partición de  $N+2$  puntos en el intervalo dominio. Se evaluarán los valores

promedio de la función  $u$  y con ellos, por medio proceso de reconstrucción usando la condición de frontera de Dirichlet para los coeficientes faltantes, reconstruimos la función  $u$ .

Las Figuras 3.2 y 3.3 muestran la reconstrucción lineal y cuadrática de la función propuesta tomando diferentes particiones. A medida que aumenta el refinamiento ( $N = 4, 8, 16$ ), se obtiene una mejor reconstrucción. Para  $N = 4$ , en ambas Figuras 3.2(a) y 3.3(b) se puede apreciar la diferencia en la reconstrucción lineal y cuadrática. De otro parte, en la Figura 3.4 se observa las curvas de convergencia para diferentes reconstrucciones. Estas curvas muestran una tendencia lineal cuyos pendientes( con valores opuestos negativos) coinciden con el orden de convergencia  $p$  ( donde el orden es  $O(h^p)$ ). Este orden de convergencia también se aprecia en el Cuadro 3.1, donde se tabula los errores de aproximación de la función  $u(x) = \cos(\pi x)$  con esquemas de reconstrucción de orden alto, donde

$$\|u - \tilde{u}\|_2 = \sqrt{\sum_{i=1}^n (u(x_i) - \tilde{u}_i)^2}$$

y  $\tilde{u}(x)$  es la función aproximada.

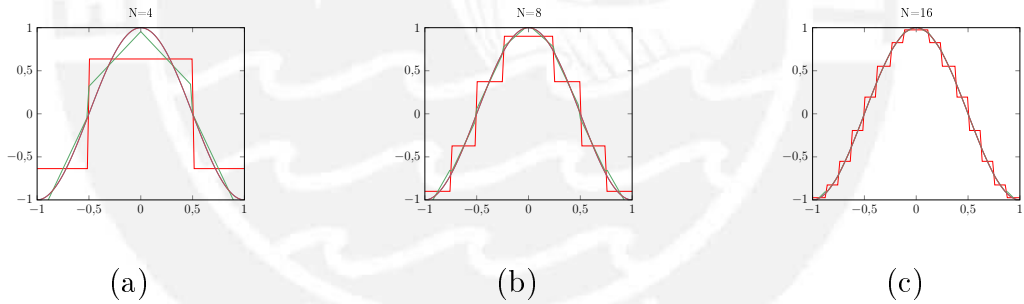


Figura 3.2: Reconstrucción lineal y valores promedio de  $u(x) = \cos(\pi x)$  en  $[-1, 1]$  con  $N = 4$ (a)  $N = 8$  (b) y  $N = 16$ (c) intervalos

En el siguiente capítulo nuestra atención se concentra en el cálculo del flujo numérico, la otra componente del MVF, y junto con la reconstrucción podemos iterar en el tiempo.

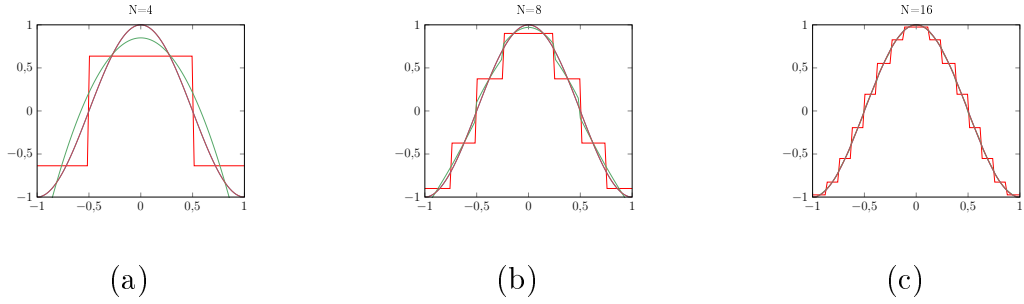


Figura 3.3: Reconstrucción cuadrática y valores promedio de  $u(x) = \cos(\pi x)$  en  $[-1, 1]$  con  $N = 4$ (a)  $N = 8$  (b) y  $N = 16$ (c) intervalos

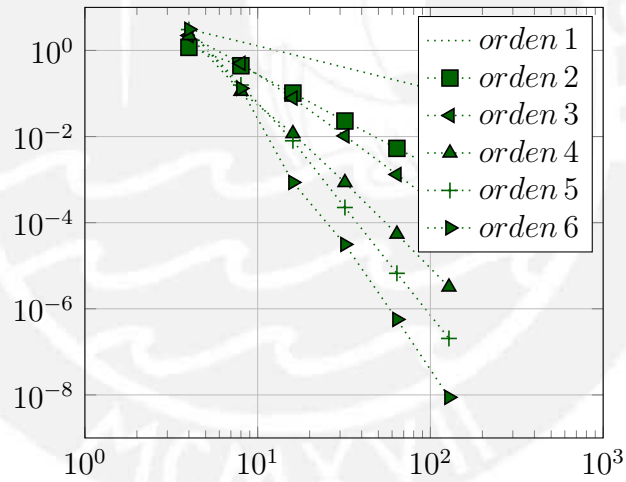


Figura 3.4: Curvas de convergencia de primer a sexto orden para una reconstrucción constante por partes ( $k = 0$ ), lineal ( $k = 1$ ), parabólico ( $k = 2$ ), cúbico ( $k = 3$ ),  $k = 4$  y  $k = 5$  para la función  $u(x) = \cos(\pi x)$  en  $[-1, 1]$  con parámetros  $w_i = 1$ .

Cuadro 3.1: Errores de aproximación de la función  $u(x) = \cos(\pi x)$  para la reconstrucción de orden alto y sus respectivos órdenes numéricos, con  $\tilde{u}(x)$  la función aproximada.

$N$	$k = 0$		$k = 1$		$k = 2$	
	$\ u - \tilde{u}\ _2$	orden	$\ u - \tilde{u}\ _2$	orden	$\ u - \tilde{u}\ _2$	orden
4	3.10E+00		1.18E+00		2.22E+00	
8	1.59E+00	0.963	4.41E-01	1.416	5.03E-01	2.144
16	8.01E-01	0.991	1.02E-01	2.107	7.81E-02	2.687
32	4.01E-01	0.998	2.30E-02	2.157	1.04E-02	2.913
64	2.01E-01	0.999	5.32E-03	2.110	1.32E-03	2.979
128	1.00E-01	1.000	1.29E-03	2.045	1.65E-04	2.995
$N$	$k = 3$		$k = 4$		$k = 5$	
	$\ u - \tilde{u}\ _2$	orden	$\ u - \tilde{u}\ _2$	orden	$\ u - \tilde{u}\ _2$	orden
4	2.22E+00		3.06E+00		3.09E+00	
8	1.13E-01	4.304	1.57E-01	4.280	1.32E-01	4.555
16	1.17E-02	3.263	7.99E-03	4.299	8.62E-04	7.254
32	8.56E-04	3.775	2.25E-04	5.148	3.13E-05	4.784
64	5.43E-05	3.980	6.67E-06	5.079	5.66E-07	5.789
128	3.21E-06	4.080	2.05E-07	5.026	8.79E-09	6.009

## Capítulo 4

### MVF Compacto: Flujo

En el capítulo anterior no se exigió continuidad entre los polinomios de reconstrucción  $u_i(x)$  obtenidos en cada intervalo  $I_i$ . Por tanto, puede existir discontinuidad en los extremos de dos intervalos consecutivos:  $x_{i\pm\frac{1}{2}}$ . Luego, evaluar el flujo numérico  $\hat{f}_{i\pm\frac{1}{2}}$  no es tan simple. En este capítulo se desea entender el cálculo numérico del flujo a través de un problema similar llamado problema de Riemman.

Supongamos que tenemos la ecuación de advección unidimensional, llamada ecuación de onda unidimensional,

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0$$

donde  $a$  es la velocidad de propagación de una onda constante. Si colocamos una condición inicial tenemos el problema de valor inicial (PVI)

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, & x \in \mathbb{R}, t > 0 \\ u(x, 0) = u_0(x). \end{cases} \quad (4.1)$$

Cuando la condición inicial es discontinua se le llama Problema de Riemann y es ampliamente estudiado por F. Toro [19].



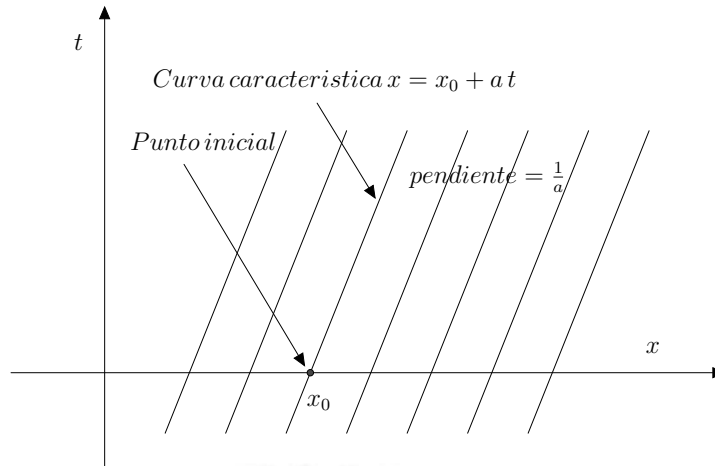


Figura 4.1: Curvas características de la ecuación de advección lineal para una velocidad característica positiva  $a$ . Condición inicial en el tiempo  $t = 0$  fija la posición inicial en  $x_0$ .

#### 4.1. Características y la solución general

Las curvas características para la ecuación (4.1) pueden ser definidas como las curvas  $x = x(t)$  en el plano  $t - x$  a lo largo del cual la EDP se convierte en una EDO. Así, si  $x(t)$  es una curva característica, donde

$$\frac{dx}{dt} = a, \quad (4.2)$$

y  $u(t, x) = u(t, x(t))$  entonces

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} \frac{dx}{dt} = \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0.$$

De este modo  $u$  permanece constante a lo largo de la curva característica  $x = x(t)$  con  $x'(t) = a$ . La velocidad  $a$  es llamada *velocidad característica* y es la pendiente de la curva  $x = x(t)$  en el plano  $t - x$ . En la bibliografía se considera el plano  $x - t$  para esbozar las curvas características; teniendo presente que sus pendientes son iguales a  $1/a$ . En la Figura 4.1 se muestra una familia de curvas características  $x = x(t)$  de la ecuación (4.2).

Así, si se da las condiciones iniciales  $u(x, 0) = u_0(x)$  en el tiempo  $t = 0$ ,

entonces a lo largo de cada curva característica  $x(t) = x_0 + at$  la solución es

$$u(x, t) = u_0(x_0) = u_0(x - at). \quad (4.3)$$

Usando el argumento geométrico anterior hemos construido la solución analítica del PVI (4.1) para la ecuación de advección y está dada en términos de  $u_0(x)$ .

## Problema de Riemann

Veamos ahora el estudio especial del PVI llamado problema de Riemann

$$\begin{cases} u_t + a u_x = 0, & x \in \mathbb{R}, t > 0 \\ u(x, 0) = u_0(x) = \begin{cases} u_L, & \text{si } x < 0, \\ u_R, & \text{si } x > 0, \end{cases} \end{cases} \quad (4.4)$$

donde  $u_L$  y  $u_R$  son dos valores constantes, como se muestra en la Figura 4.2. Note que el valor inicial tiene una discontinuidad en  $x = 0$ . La solución del PVI (4.4) es (similar a la solución dada en la ecuación (4.3))

$$u(x, t) = u_0(x_0) = u_0(x - at) = \begin{cases} u_L, & \text{si } x - at < 0, \\ u_R, & \text{si } x - at > 0. \end{cases} \quad (4.5)$$

La solución del problema de Riemann representado en el plano  $x-t$  es mostrado en la Figura 4.3.

## 4.2. Elección del flujo numérico

Debido al proceso de reconstrucción se tiene un problema similar al problema de Riemann en cada una de las celdas  $I_i$  como se muestra en la Figura 4.4. En ella se tiene que discernir el flujo  $\hat{f}$  en los extremos del intervalo  $I_i$ . Por ejemplo, en  $x_{i+\frac{1}{2}}$  tenemos por el lado izquierdo  $u_{i+\frac{1}{2}}^L = u_i \left( x_{i+\frac{1}{2}} \right)$  y por el lado derecho

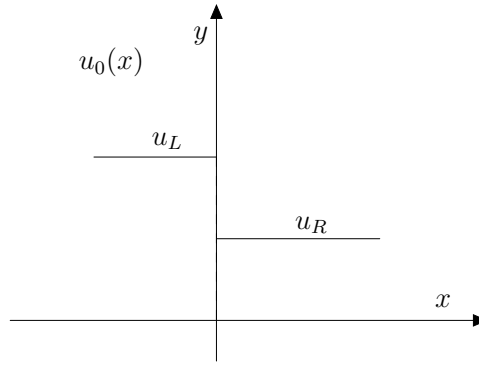


Figura 4.2: Condición inicial para un problema de Riemman. En el tiempo inicial el dato consiste de dos constantes de estado separadas por una discontinuidad en  $x = 0$

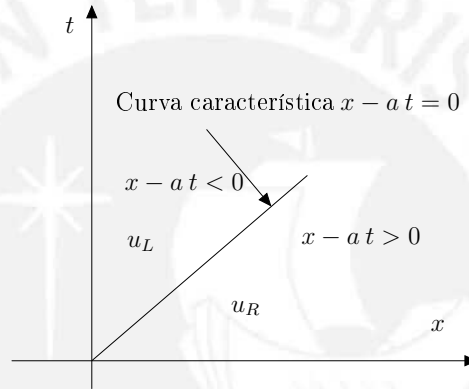


Figura 4.3: Ilustración de la solución del problema de Riemman en el plano  $x - t$  para la ecuación de advección lineal con velocidad de característica positiva  $a$ .

$u_{i+\frac{1}{2}}^R = u_{i+1}\left(x_{i+\frac{1}{2}}\right)$ . La pregunta es: ¿cuál debo usar para definir el flujo  $\hat{f}_{i+\frac{1}{2}}$ ,  $u_{i+\frac{1}{2}}^L$  o  $u_{i+\frac{1}{2}}^R$ ?

Recordemos que la ecuación (2.5) está dada por

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right),$$

donde  $f_{i\pm\frac{1}{2}} = f\left(u\left(t, x_{i\pm\frac{1}{2}}\right)\right) = au\left(t, x_{i\pm\frac{1}{2}}\right)$  son los flujos exactos en cada volumen de control  $I_i$ . Los métodos para determinar el flujo numérico se desarrollan ampliamente en [19] y en [2]. Chi-Wan en [17] plantea las siguientes condiciones para la función  $\hat{f}$ :

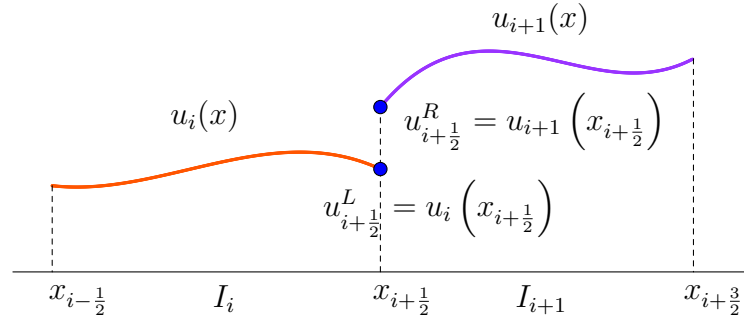


Figura 4.4: Elección de flujo entre dos polinomios de reconstrucción  $u_i(x)$  y  $u_{i+1}(x)$ .

- $\hat{f}(u^L, u^R)$  es una función continua Lipschitz en ambos argumentos;
- $\hat{f}(u^L, u^R)$  es una función no decreciente en  $u^L$  y una función no creciente en  $u^R$ . Simbólicamente  $\hat{f}(\uparrow, \downarrow)$ ;
- $\hat{f}(u^L, u^R)$  es consistente con el flujo físico; es decir,  $\hat{f}(u, u) = f(u)$ .

Si un flujo numérico  $\hat{f}$  las satisface, entonces es llamado flujo monótono. Algunos flujos monótonos que satisfacen estas condiciones son los siguientes:

1. Flujo de Godunov

$$\hat{f}(x, y) = \begin{cases} \min_{x \leq u \leq y} f(u), & \text{si } x \leq y; \\ \max_{x \leq u \leq y} f(u), & \text{si } x > y \end{cases};$$

2. Flujo de Engquist–Osher

$$\hat{f}(x, y) = \int_0^x \max\{f'(u), 0\} du + \int_0^y \max\{f'(u), 0\} du + f(0);$$

3. Flujo de Lax–Friedrichs

$$\hat{f}(x, y) = \frac{1}{2} [f(x) + f(y) - \alpha(y - x)],$$

donde  $\alpha = \max_u f'(u)$  es una constante.

De acuerdo a estos flujos, como para la ecuación de advección unidimensional se tiene  $f(u) = au$ , entonces  $f'(u) = a$ . Luego, por ejemplo, para el flujo de Lax–Friedrichs sería

$$\hat{f}(x, y) = \frac{1}{2} [ax + ay - a(y - x)] = ax,$$

el cual será usado más adelante.

### 4.3. Esquema de volúmenes finitos para la ecuación de advección lineal

En esta sección usaremos los esquemas de volúmenes finitos para aproximar la solución a una ecuación de advección lineal escalar unidimensional. La ecuación es

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad (4.6)$$

donde el flujo es  $f(u(t, x)) = f(u) = au$  y  $a$  es una velocidad de propagación, la cual asumiremos positiva, i.e.,  $a > 0$ . También asumiremos una discretización uniforme del dominio espacial con  $\Delta x = h$ . La forma integral de la ecuación (4.6) sobre el volumen de control  $I_i$  es

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right), \quad (4.7)$$

donde  $f_{i\pm\frac{1}{2}} = f\left(u\left(t, x_{i\pm\frac{1}{2}}\right)\right) = au\left(t, x_{i\pm\frac{1}{2}}\right)$  son los flujos exactos.

Un método de volúmenes finitos proporciona la forma semi discreta de la ecuación (4.7) introduciendo un flujo numérico  $\hat{f}_{i\pm\frac{1}{2}}$

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right). \quad (4.8)$$

Cuando se tiene un flujo de avance, el **flujo numérico** de Lax–Friedrichs (Ver Sección 4.2) es calculado por

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}\left(u_{i+\frac{1}{2}}^L; u_{i+\frac{1}{2}}^R\right) = au_{i+\frac{1}{2}}^L \quad (4.9)$$

para  $a > 0$ , donde los valores de  $u_{i+\frac{1}{2}}^L$  y  $u_{i+\frac{1}{2}}^R$  son calculados por el polinomio reconstruido sobre  $I_i$  y  $I_{i+1}$  (Ver Figura 4.4),

$$\begin{aligned} u_{i+\frac{1}{2}}^L &= u_i \left( x_{i+\frac{1}{2}} \right) = \bar{u}_i + \sum_{l=1}^k u_i^l \varphi_{l,i} \left( x_{i+\frac{1}{2}} \right), \\ u_{i+\frac{1}{2}}^R &= u_{i+1} \left( x_{i+\frac{1}{2}} \right) = \bar{u}_{i+1} + \sum_{l=1}^k u_{i+1}^l \varphi_{l,i+1} \left( x_{i+\frac{1}{2}} \right), \end{aligned} \quad (4.10)$$

donde los coeficientes  $u_i^l$  y  $u_{i+1}^l$  son determinados por el procedimiento de reconstrucción de la Sección (3.1). De manera similar para  $\hat{f}_{i-\frac{1}{2}}$ .

En caso usemos el método de Euler de un paso para la discretización de la Ecuación (4.8) tenemos la forma discreta

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\Delta t}{\Delta x} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right). \quad (4.11)$$

Para los cálculos numéricos del Capítulo 6 usaremos los métodos de Runge-Kutta mencionados en la Sección 2.3.

Luego de estudiar las dos componentes más importantes de los métodos de volúmenes finitos, afrontaremos en el siguiente capítulo, con ayuda del análisis de Fourier, el análisis espectral, tanto de los esquemas de reconstrucción, como de la estabilidad de los esquemas de evolución en el tiempo.

# Capítulo 5

## Análisis de Fourier

En este capítulo desarrollamos los elementos más importantes del análisis espectral basado en los estudios de Vichnewetsky [21] donde se muestra el uso sistemático del análisis de Fourier para investigar problemas relacionados a los errores en aproximaciones numéricas en las ecuaciones hiperbólicas. De este modo, mostraremos el comportamiento espectral de esquemas CLSFV, de segundo a cuarto orden, los que nos permitirá establecer parámetros optimales. Así mismo, hacemos uso del análisis de von Neumann para estudiar la estabilidad de estos esquemas.

### 5.1. Análisis espectral

Con el método de volúmenes finitos realizaremos una semi discretización para la variable espacial, para analizar la evolución temporal del error para el modelo unidimensional de la ecuación de advección

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad (5.1)$$

donde el flujo es  $f = a u$  y  $a$  es una constante.

Recordemos la discretización vista en la sección 2.2. Consideremos una partición  $\{I_i\}_{i=1}^N$  uniforme de tamaño  $N$  del dominio computacional  $\Omega$ , con  $\Delta x_i = h$  y donde los  $x_i$  son los centroides de los intervalos  $I_i = \left[ x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right]$ .

La ecuación integral exacta en el intervalo  $I_i$  esta dada por

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right) \quad (5.2)$$

y  $f_{i\pm\frac{1}{2}} = a u(x_{i\pm\frac{1}{2}}, t)$ .

Para propósitos del análisis de Fourier supongamos que la función  $u$  tiene dominio computacional  $[0, L]$  y es periódica con periodo  $L$ . Entonces, ésta puede ser descompuesta en sus coeficientes de Fourier

$$u(x, t) = \sum_{m=-\infty}^{\infty} \hat{A}_m(t) \exp\left(\frac{2\pi i m x}{L}\right)$$

donde  $i = \sqrt{-1}$ .

Ahora por simplicidad analizamos un modo de Fourier simple de la forma

$$u(x, t) = A_m(t) e^{i k_m x}, \quad k_m = \frac{2\pi m}{L}$$

como una solución de la ecuación integral exacta (5.2), donde  $A_m$  representa la amplitud del modo y  $k_m \in \mathbb{R}$  es el **número de onda**. Su valor promedio en el intervalo  $I_i$  es

$$\begin{aligned} \bar{u}_i &= \frac{1}{\Delta x_i} \int_{I_i} u(x, t) dx \\ &= \frac{(i k_m)^{-1}}{h} A_m(t) e^{i k_m x} \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \frac{A_m(t)}{i k_m h} \left( e^{i k_m x_{i+\frac{1}{2}}} - e^{i k_m x_{i-\frac{1}{2}}} \right). \end{aligned} \quad (5.3)$$

Los flujos exactos en la ecuación (2.5) son

$$f_{i\pm\frac{1}{2}} = a u(x_{i\pm\frac{1}{2}}, t) = a A_m(t) e^{i k_m x_{i\pm\frac{1}{2}}}. \quad (5.4)$$

Así, la ecuación integral exacta (5.2) esta dada por

$$\begin{aligned} \frac{d}{dt} \left( \frac{A_m(t)}{i k_m h} \left( e^{i k_m x_{i+\frac{1}{2}}} - e^{i k_m x_{i-\frac{1}{2}}} \right) \right) &= -\frac{1}{h} \left( a A_m(t) e^{i k_m x_{i+\frac{1}{2}}} - a A_m(t) e^{i k_m x_{i-\frac{1}{2}}} \right) \\ \frac{1}{i k_m h} \left( e^{i k_m x_{i+\frac{1}{2}}} - e^{i k_m x_{i-\frac{1}{2}}} \right) \frac{d}{dt} A_m(t) &= -\frac{1}{h} a A_m(t) \left( e^{i k_m x_{i+\frac{1}{2}}} - e^{i k_m x_{i-\frac{1}{2}}} \right) \end{aligned} \quad (5.5)$$



$$\frac{1}{ik_m} \frac{d}{dt} A_m(t) = -a A_m(t).$$

Entonces,

$$\frac{d}{dt} A_m(t) + a i k_m A_m(t) = 0. \quad (5.6)$$

Por otro lado de acuerdo a la ecuación (2.6), para el esquema semi discreto de la ecuación (5.2) se tiene

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right). \quad (5.7)$$

Los diferencia de los flujos exactos son calculados como

$$f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} = a u(x_{i+1/2}, t) - a u(x_{i-1/2}, t) = a A_m(t) \left( e^{ik_m x_{i+\frac{1}{2}}} - e^{ik_m x_{i-\frac{1}{2}}} \right); \quad (5.8)$$

o sea,

$$e^{ik_m x_{i+\frac{1}{2}}} - e^{ik_m x_{i-\frac{1}{2}}} = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{a A_m(t)}. \quad (5.9)$$

Ahora en la ecuación (5.7) reemplazamos la expresión (5.3). Luego

$$\frac{1}{ik_m h} \left( e^{ik_m x_{i+\frac{1}{2}}} - e^{ik_m x_{i-\frac{1}{2}}} \right) \frac{d}{dt} A_m(t) = -\frac{1}{h} \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right)$$

y usando la ecuación (5.9), la ecuación anterior es equivalente a

$$\frac{1}{ik_m} \left( f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right) \frac{d}{dt} A_m(t) = -a A_m(t) \left( \hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right);$$

de modo que podemos expresarlo como

$$\frac{d}{dt} A_m(t) + a k'_m i A_m(t) = 0, \quad (5.10)$$

donde

$$\frac{k'_m}{k_m} = \frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}.$$

Los valores de  $\hat{f}_{i\pm\frac{1}{2}}$  son evaluados usando la ecuación (4.9); es decir,

$$\hat{f}_{i+\frac{1}{2}} = a u_{i+\frac{1}{2}}^L = a u_i \left( x_{i+\frac{1}{2}} \right), \hat{f}_{i-\frac{1}{2}} = a u_{i-\frac{1}{2}}^L = a u_{i-1} \left( x_{i-\frac{1}{2}} \right),$$

donde los  $u_i \left( x_{i+\frac{1}{2}} \right)$  y  $u_{i-1} \left( x_{i-\frac{1}{2}} \right)$  son calculados usando la ecuación (4.10) con los coeficientes obtenidos en el proceso de reconstrucción y los valores medios de la ecuación (5.3). Para poder determinar estos valores introducimos un número de onda escalado

$$\kappa = k_m h \in (0, \pi];$$

de modo que  $\kappa' = k'_m h$  es expresado por

$$\kappa' = \frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}} \kappa.$$

$\kappa'$  es llamado **número de onda modificado** cuya parte real  $Re(\kappa')$  y su parte imaginaria  $Im(\kappa')$  son asociados a propiedades de dispersión y disipación de los esquemas numéricos respectivamente. La parte real describen los errores de fase y las partes imaginarias describen los errores de amplitud disipativa. Un esquema con ningún error debería satisfacer  $\kappa' = \kappa$ , para  $0 \leq \kappa \leq \pi$ ; es decir, la parte imaginaria debería ser cero.

## 5.2. Cálculo de los números de onda modificados

En esta sección presentamos los números de onda modificados  $\kappa'$  de segundo, tercer y cuarto orden para esquemas CLSFV. Para mostrar la dificultad en el cálculo presentamos el desarrollo del esquema de segundo orden. Para los órdenes superiores los resultados obtenidos fueron procesados usando programación simbólica.

### 5.2.1. Número de onda modificado para un esquema de segundo orden ( $k = 1$ )

Para un esquema de segundo orden ( $k = 1$ ), teniendo en cuenta la Figura 5.1, los polinomios base dados en la Ecuación (3.4) vistos en la sección 3.1 son los

siguientes:

$$\varphi_{1,i}(x) = \frac{x - x_i}{h} - 0 = \frac{x - x_i}{h};$$

de modo que  $\varphi_{1,i}\left(x_{i+\frac{1}{2}}\right) = \frac{1}{2}$  y  $\varphi_{1,i-1}\left(x_{i-\frac{1}{2}}\right) = \frac{1}{2}$ .

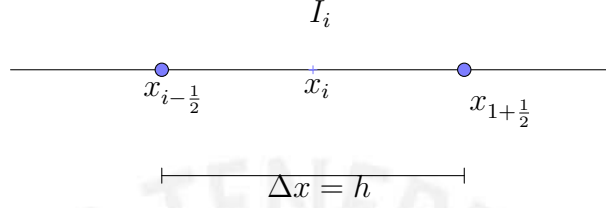


Figura 5.1: Longitud de una celda  $I_i$

De otro lado, para el mismo  $k = 1$  los flujos numéricos  $\hat{f}_{i\pm\frac{1}{2}}$  son obtenidos como sigue:

$$\begin{aligned}\hat{f}_{i+\frac{1}{2}} &= a u_i \left(x_{i+\frac{1}{2}}\right) = a \left( \bar{u}_i + \sum_{l=1}^{k=1} u_i^l \varphi_{l,i} \left(x_{i+\frac{1}{2}}\right) \right) \\ &= a \left( \bar{u}_i + u_i^1 \varphi_{1,i} \left(x_{i+\frac{1}{2}}\right) \right).\end{aligned}$$

Los valores promedios  $\bar{u}_i$  son obtenidos usando la ecuación (5.3) y los coeficientes  $u_i^1$  por medio de la solución del sistema de ecuaciones (3.18) del siguiente modo. Considerando que para  $k = 1$  y la ecuación (3.24), los coeficientes matriciales en (3.18) se reducen a  $\mathbf{E}_i = \mathbf{0} = \mathbf{F}_i$ ,  $\mathbf{D}_i = \mathbf{A}_i^T \mathbf{A}_i = 2$ ,  $\mathbf{g}_i = \mathbf{A}_i \mathbf{b}_i = \bar{u}_{i+1} - \bar{u}_{i-1}$ . Entonces, el sistema

$$\mathbf{E}_i \mathbf{u}_{i-1} + \mathbf{D}_i \mathbf{u}_i + \mathbf{F}_i \mathbf{u}_{i+1} = \mathbf{g}_i, \quad i = 2, \dots, N-1$$

tiene solución

$$u_i^1 = \frac{\bar{u}_{i+1} - \bar{u}_{i-1}}{2}$$

$$\begin{aligned}
&= \frac{1}{2} \left( \frac{A_m(t)}{ik_m h} \left( e^{ik_m x_{i+\frac{3}{2}}} - e^{ik_m x_{i+\frac{1}{2}}} \right) - \frac{A_m(t)}{ik_m h} \left( e^{ik_m x_{i-\frac{1}{2}}} - e^{ik_m x_{i-\frac{3}{2}}} \right) \right) \\
&= \frac{A_m(t)}{2ik_m h} e^{ik_m x_i} \left( e^{ik_m \frac{3}{2}h} - e^{ik_m \frac{h}{2}} - e^{-ik_m \frac{h}{2}} + e^{-ik_m \frac{3}{2}h} \right),
\end{aligned}$$

e introduciendo el número de onda escalado  $\kappa = k_m h$  tenemos

$$u_i^1 = \frac{A_m(t)}{2ik_m h} e^{ik_m x_i} \left( e^{i\frac{3}{2}\kappa} - e^{i\frac{\kappa}{2}} - e^{-i\frac{\kappa}{2}} + e^{-i\frac{3}{2}\kappa} \right).$$

De manera similar,

$$\begin{aligned}
u_{i-1}^1 &= \frac{\bar{u}_i - \bar{u}_{i-2}}{2} \\
&= \frac{A_m(t)}{2ik_m h} e^{ik_m x_i} \left( e^{ik_m \frac{h}{2}} - e^{-ik_m \frac{h}{2}} - e^{-ik_m \frac{3h}{2}} + e^{-ik_m \frac{5h}{2}} \right) \\
&= \frac{A_m(t)}{2ik_m h} e^{ik_m x_i} \left( e^{i\frac{\kappa}{2}} - e^{-i\frac{\kappa}{2}} - e^{-i\frac{3\kappa}{2}} + e^{-i\frac{5\kappa}{2}} \right).
\end{aligned}$$

Luego, el flujo numérico es

$$\begin{aligned}
\hat{f}_{i+\frac{1}{2}} &= a \left( \bar{u}_i + u_i^1 \varphi_{1,i} \left( x_{i+\frac{1}{2}} \right) \right) \\
&= a \left( \bar{u}_i + u_i^1 \cdot \frac{1}{2} \right) \\
&= \frac{aA_m(t)}{ik_m h} e^{ik_m x_i} \left( e^{ik_m \frac{h}{2}} - e^{-ik_m \frac{h}{2}} \right) + \frac{aA_m(t)}{4ik_m h} e^{ik_m x_i} \left( e^{ik_m \frac{3}{2}h} - e^{ik_m \frac{h}{2}} - e^{-ik_m \frac{h}{2}} + e^{-ik_m \frac{3}{2}h} \right) \\
&= \frac{aA_m(t)}{ik_m h} e^{ik_m x_i} \left( \frac{3}{4} e^{ik_m \frac{h}{2}} - \frac{5}{4} e^{-ik_m \frac{h}{2}} + \frac{1}{4} e^{ik_m \frac{3}{2}h} + \frac{1}{4} e^{-ik_m \frac{3}{2}h} \right) \\
&= \frac{aA_m(t)}{ik_m h} e^{ik_m x_i} \left( \frac{3}{4} e^{i\frac{\kappa}{2}} - \frac{5}{4} e^{-i\frac{\kappa}{2}} + \frac{1}{4} e^{i\frac{3}{2}\kappa} + \frac{1}{4} e^{-i\frac{3}{2}\kappa} \right)
\end{aligned}$$

y para

$$\begin{aligned}
\hat{f}_{i-\frac{1}{2}} &= a \left( \bar{u}_{i-1} + u_{i-1}^1 \varphi_{1,i} \left( x_{i-\frac{1}{2}} \right) \right) \\
&= a \left( \bar{u}_{i-1} + u_{i-1}^1 \cdot \frac{1}{2} \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{aA_m(t)}{ik_m h} e^{ik_m x_i} \left( e^{-i\frac{\kappa}{2}} - e^{-i\frac{3\kappa}{2}} \right) + \frac{aA_m(t)}{4ik_m h} e^{ik_m x_i} \left( e^{i\frac{\kappa}{2}} - e^{-i\frac{\kappa}{2}} - e^{-i\frac{3\kappa}{2}} + e^{-i\frac{5\kappa}{2}} \right) \\
&= \frac{aA_m(t)}{ik_m h} e^{ik_m x_i} \left( \frac{1}{4} e^{i\frac{\kappa}{2}} - \frac{3}{4} e^{-i\frac{\kappa}{2}} - \frac{5}{4} e^{-i\frac{3\kappa}{2}} + \frac{1}{4} e^{-i\frac{5\kappa}{2}} \right)
\end{aligned}$$

De este modo,

$$\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} = \frac{aA_m(t)}{i\kappa} e^{ik_m x_i} \left( \frac{1}{2} e^{\frac{\kappa}{2}} - 2e^{-\frac{i\kappa}{2}} + \frac{1}{4} e^{i\frac{3}{2}\kappa} + \frac{3}{2} e^{-i\frac{3}{2}\kappa} - \frac{1}{4} e^{-i\frac{5\kappa}{2}} \right)$$

y

$$\begin{aligned}
f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} &= aA_m(t) e^{ik_m x_i} \left( e^{ik_m \frac{h}{2}} - e^{-ik_m \frac{h}{2}} \right) \\
&= aA_m(t) e^{ik_m x_i} \left( e^{i\frac{\kappa}{2}} - e^{-i\frac{\kappa}{2}} \right).
\end{aligned}$$

De allí que

$$\begin{aligned}
\kappa' &= \frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}} \kappa \\
&= \frac{-i}{\left( e^{i\frac{\kappa}{2}} - e^{-i\frac{\kappa}{2}} \right)} \left( \frac{1}{2} e^{\frac{\kappa}{2}} - 2e^{-\frac{i\kappa}{2}} + \frac{1}{4} e^{i\frac{3}{2}\kappa} + \frac{3}{2} e^{-i\frac{3}{2}\kappa} - \frac{1}{4} e^{-i\frac{5\kappa}{2}} \right) \\
&= -i \left( \frac{1}{2} - \frac{3}{2} e^{-i\kappa} + \frac{1}{4} e^{i\kappa} (1 + e^{-2\kappa i}) (1 + e^{-\kappa i}) \right) \\
&= -i \left( \frac{3}{4} - \frac{5}{4} \cos(\kappa) + \frac{5}{4} \sin(\kappa) i + \frac{1}{4} \cos(\kappa) + i \frac{1}{4} \sin(\kappa) + \frac{1}{4} \cos(2\kappa) \right) \\
&= \frac{3}{2} \sin(\kappa) - \frac{1}{4} \sin(2\kappa) + i \left( -\frac{3}{4} + \cos(\kappa) - \frac{1}{4} \cos(2\kappa) \right).
\end{aligned}$$

Finalmente, para un esquema de segundo orden las propiedades de dispersión y disipación están dados por

$$\begin{aligned}
Re(\kappa') &= \frac{3}{2} \sin(\kappa) - \frac{1}{4} \sin(2\kappa), \\
Im(\kappa') &= -\frac{3}{4} + \cos(\kappa) - \frac{1}{4} \cos(2\kappa),
\end{aligned} \tag{5.11}$$

respectivamente.

### 5.2.2. Número de onda modificado para esquemas de órdenes superiores

Para cada uno de los esquemas de segundo a cuarto orden se tuvo que recurrir a la programación simbólica siguiendo los pasos de la sección anterior, los resultados se muestran a continuación:

#### Tercer orden ( $k = 2$ )

$$Re(\kappa') = \frac{1}{4w_1^4 + 5w_1^2 + 1} \left( -4w_1^4 \sin(K) \cos(2K) + 4w_1^4 \sin(2K) \cos(K) - \frac{67}{12} w_1^2 \sin(K) \cos(2K) + \right. \\ \left. + \frac{67}{12} w_1^2 \sin(2K) \cos(K) - \frac{11}{12} w_1^2 \sin(2K) \cos(3K) - \frac{13}{12} w_1^2 \sin(2K) + \frac{11}{12} w_1^2 \sin(3K) \cos(2K) + \right. \\ \left. - \sin(K) \cos(2K) + \sin(2K) \cos(K) - \frac{1}{3} \sin(2K) \cos(3K) + -\frac{1}{6} \sin(2K) + \frac{1}{3} \sin(3K) \cos(2K) \right),$$

$$Im(\kappa') = \frac{1}{4w_1^4 + 5w_1^2 + 1} \left( 4w_1^4 \sin(\kappa) \sin(2\kappa) - 4w_1^4 \sin(2\kappa)^2 + 4w_1^4 \cos(\kappa) \cos(2\kappa) - 4w_1^4 \cos(2\kappa)^2 + \right. \\ \left. + \frac{67}{12} w_1^2 \sin(\kappa) \sin(2\kappa) - \frac{43}{12} w_1^2 \sin(2\kappa)^2 - \frac{11}{12} w_1^2 \sin(2\kappa) \sin(3\kappa) \right. \\ \left. + \frac{67}{12} w_1^2 \cos(\kappa) \cos(2\kappa) - \frac{43}{12} w_1^2 \cos(2\kappa)^2 - \frac{11}{12} w_1^2 \cos(2\kappa) \cos(3\kappa) \right. \\ \left. - \frac{13}{12} w_1^2 \cos(2\kappa) + \sin(\kappa) \sin(2\kappa) - \frac{1}{2} \sin(2\kappa)^2 - \frac{1}{3} \sin(2\kappa) \sin(3\kappa) \right. \\ \left. + \cos(\kappa) \cos(2\kappa) - \frac{1}{2} \cos(2\kappa)^2 - \frac{1}{3} \cos(2\kappa) \cos(3\kappa) - \frac{1}{6} \cos(2\kappa) \right). \quad (5.12)$$

#### Cuarto orden ( $k = 3$ )

Definiendo

$$s_1 = w_1^2(77 + 318w_1^2) + 6(111 + 538w_1^2 + 180w_1^4)w_2^2 + 216(13 + 10w_1^2)w_2^4 \\ s_2 = -4(w_1^2 + 6w_1^4) + 6(6 + 41w_1^2 + 36w_1^4)w_2^2 + 432(1 + w_1^2)w_2^4$$

$$\begin{aligned}
s_3 &= (1 + 6w_1^2 + 12w_2^2)(w_1^2 + 18(1 + 2w_1^2)w_2^2) \\
s_4 &= 4(1 + 6w_1^2 + 12w_2^2)(w_1^2 + 18(1 + 2w_1^2) + w_2^2)
\end{aligned}$$

y,

$$\begin{aligned}
c_0 &= 48(w_1^2 + 4w_1^4) + 432(1 + 5w_1^2 + 3w_1^4)w_2^2 + 864(2 + 3w_1^2)w_2^4 \\
c_1 &= -24(72w_2^4 + w_1^4(-4 + 72w_2^2) + w_1^2(-1 + 22w_2^2 + 144w_2^4)) \\
c_2 &= 48w_1^2w_2^2(2 + 9w_1^2 + 18w_2^2) \\
c_3 &= 6w_1^2 + 24w_1^4 + 54(1 + 5w_1^2 + 3w_1^4)w_2^2 + 108(2 + 3w_1^2)w_2^4 \\
c_4 &= -216w_2^4 - 3w_1^4(-4 + 72w_2^2) - 3w_1^2(-1 + 22w_2^2 + 144w_2^4)c_5 = 6w_1^2w_2^2(2 + 9w_1^2 + 18w_2^2);
\end{aligned}$$

entonces,

$$\begin{aligned}
Re(\kappa') &= \frac{s_1 \text{sen}(\kappa) + s_2 \text{sen}(2\kappa) + s_3 \text{sen}(3\kappa)}{c_0 + c_1 \cos(\kappa) + c_2 \cos(2\kappa)}, \\
Im(\kappa') &= \frac{-s_4 (\text{sen}(\kappa/2))^6}{c_3 + c_4 \cos(\kappa) + c_5 \cos(2\kappa)}.
\end{aligned} \tag{5.13}$$

De acuerdo a las ecuaciones (5.12) y (5.13), tenemos un parámetro libre  $w_1$  en el esquema de tercer orden y dos parámetros  $(w_1, w_2)$  en el esquema de cuarto orden.

### 5.3. Optimización de parámetros

Wang [22] propone un proceso de optimización para la elección de los parámetros libres  $w_i$  en base a lo que él llama función costo [8]

$$E = \frac{1}{e^{\nu\pi}} \int_0^\pi e^{\nu(\pi-k)} (Re(\kappa') - \kappa)^2 d\kappa,$$

que no es otra cosa que una función error integral del número de onda modificado. Los valores optimizados de  $(w_1, w_2)$  para diferentes valores  $\nu$  se han tomado de la Tabla 1 de Wang [22].

$\nu$	2	4	6	8	10	20	40	60	80
$w_1$	$>10^4$	1	$<10^{-4}$	0,020	0,0100	0,0100	0,0100	0,0100	0,0100
$w_2$	0	0	0	0,0360	0,0490	0,0196	0,0144	0,0121	0,0100

Cuadro 5.1: Valores optimizados de  $(w_1, w_2)$  para diferentes valores de  $\nu$  (Tomado de [22])

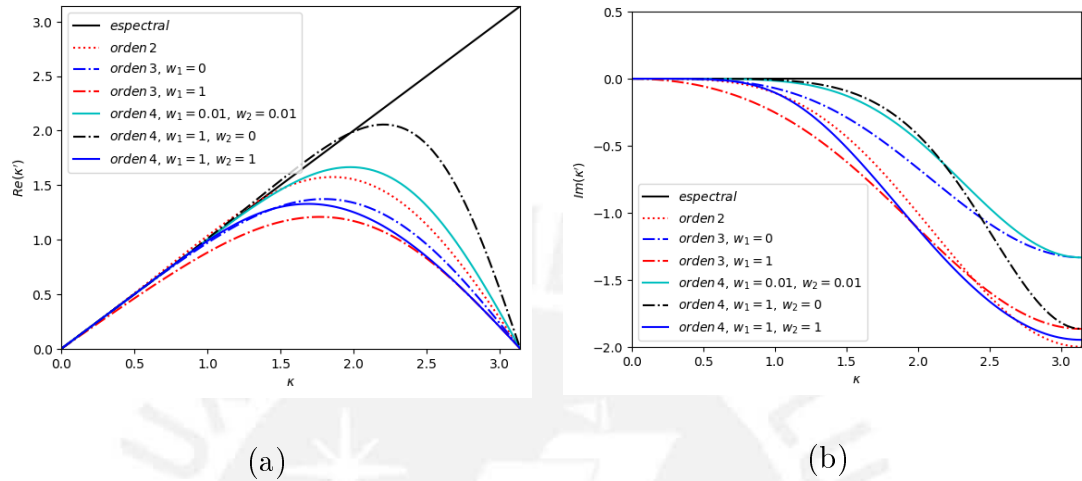


Figura 5.2: Propiedades de dispersión (a) y disipación (b) de esquemas CLSFV de segundo a cuarto orden

Las Figuras 5.2 (a) y 5.2 (b) muestran las propiedades de dispersión ( $Re(\kappa')$ ) y disipación ( $Im(\kappa')$ ) para diferentes números de onda en los esquemas CLSFV de segundo a cuarto orden. La parte real y parte imaginaria casi coinciden con los valores exactos para valores bajos de  $\kappa$ , pero para números de onda altos la coincidencia es pobre. Si la parte imaginaria de  $\kappa'$  es negativa, el esquema es disipativo, y como se observa para número de onda alto los esquemas se muestran disipativos.

En especial para los esquemas de tercer y cuarto orden se muestran las propiedades de dispersión y disipación correspondientes a parámetros optimales en el Cuadro 5.1. Para los esquemas de cuarto orden, los errores de dispersión son más pequeños para los parámetros óptimos  $w_1 = 0,01$  y  $w_2 = 0,01$  para números de onda bajos, mientras que para los valores de los parámetros  $w_1 = 1$  y  $w_2 = 0$  se tiene menor error de dispersión para números de onda altos. Siguiendo las observación de Wang [22] para comparar mejor estos resultados podemos ver en la



Figura 5.3 los comportamientos relativos a las propiedades de dispersión comparando  $\frac{Re(\kappa')}{\kappa} - 1$  con 0. Para  $(w_1 = 0,01, w_2 = 0,01)$ , el ancho de banda con error de fase relativo  $\left| \frac{Re(\kappa')}{\kappa} - 1 \right| \leq 0,5\%$  ocurre cuando  $\kappa \in (0; 1,23]$ , mientras que un error relativo menor al 3% ocurre cuando  $\kappa \in (0; 1,51]$ . Para  $(w_1 = 1, w_2 = 0)$ , el ancho de banda con error de fase relativo  $\left| \frac{Re(\kappa')}{\kappa} - 1 \right| \leq 0,5\%$  ocurre cuando  $\kappa \in (0; 0,82]$  y el ancho de banda con error de fase relativo  $\left| \frac{Re(\kappa')}{\kappa} - 1 \right| \leq 3\%$  ocurre con  $\kappa \in (0; 2,10]$ . Para ambos casos, el esquema de cuarto orden produce mejores resultados de ancho de banda que los esquemas de segundo y tercer orden para la misma tolerancia del error de fase.

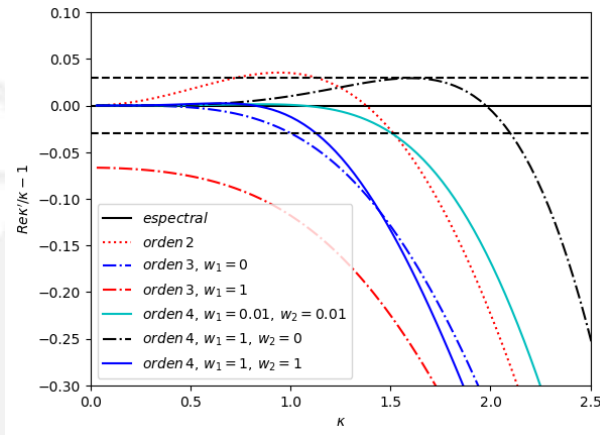


Figura 5.3: Error de dispersión para esquemas CLSFV de segundo a cuarto orden con un ancho de banda del 3%.

## 5.4. Análisis de estabilidad de von Neumann

En esta sección veremos cómo se comporta la estabilidad de von Neumann con la combinación del MVF estudiados con dos métodos de RK. Reescribimos la Ecuación (5.10) de la siguiente forma:

$$\frac{dA_m}{dt} = -a i k'_m A_m. \quad (5.14)$$

### 5.4.1. Análisis con Runge-Kutta 3

Para la ecuación (5.14) usamos un Runge-Kutta 3 visto en la sección 2.3 y  $c = a\Delta t/h$  el número CFL, tenemos

$$\begin{aligned} A_m^{(1)} &= A_m^n - ai\frac{\kappa'}{h}\Delta t A_m^n = (1 - i\kappa'c) A_m^n \\ A_m^{(2)} &= \frac{3}{4}A_m^n + \frac{1}{4}A_m^{(1)} - \frac{1}{4}i\kappa'c A_m^{(1)} \\ &= \frac{3}{4}A_m^n + \frac{1}{4}(1 - i\kappa'c)^2 A_m^n \\ &= A_m^n \left( \frac{3}{4} + \frac{1}{4}(1 - i\kappa'c)^2 \right). \end{aligned}$$

Ahora,

$$\begin{aligned} A_m^{(n+1)} &= \frac{1}{3}A_m^n + \frac{2}{3}A_m^{(2)} - \frac{2}{3}\Delta t G(A_m^{(2)}) \\ &= \frac{1}{3}A_m^n + \frac{2}{3}A_m^{(2)}(1 - i\kappa'c) \\ &= \frac{1}{3}A_m^n + \frac{2}{3}(1 - i\kappa'c) \left( \frac{3}{4} + \frac{1}{4}(1 - i\kappa'c)^2 \right) A_m^n \\ &= A_m^n \left[ \frac{1}{3} + \frac{2}{3}(1 - i\kappa'c) \left( \frac{3}{4} + \frac{1}{4}(1 - i\kappa'c)^2 \right) \right] \\ &= A_m^n \left[ \frac{1}{3} + \frac{2}{3}(1 + z) \left( \frac{3}{4} + \frac{1}{4}(1 + z)^2 \right) \right] \\ &= A_m^n \left[ \frac{1}{3} + \frac{1}{2}(1 + z) + \frac{1}{6}(1 + z)^3 \right], \end{aligned}$$

haciendo  $z = -i\kappa'c$ . Entonces, el factor de amplificación puede ser obtenido por el cociente

$$G_m = \frac{A_m^{n+1}}{A_m^n} = \frac{1}{3} + \frac{1}{2}(1 + z) + \frac{1}{6}(1 + z)^3 = 1 + z + \frac{z^2}{2} + \frac{z^3}{6}.$$

### 5.4.2. Análisis con Runge-Kutta 4

Para ver la estabilidad del método de RK4 de cuatro pasos descrito en la Sección 2.3 procedemos de manera similar.

$$\begin{aligned}
A_m^{(1)} &= A_m^n - ai\frac{\kappa'}{h}\frac{\Delta t}{2}A_m^n = \left(1 - \frac{i\kappa'c}{2}\right)A_m^n. \\
A_m^{(2)} &= A_m^n - \frac{1}{2}i\kappa'cA_m^{(1)} \\
&= A_m^n - \frac{1}{2}i\kappa'c\left(1 - \frac{i\kappa'c}{2}\right)A_m^n \\
&= A_m^n\left(1 - \frac{1}{2}i\kappa'c + \left(\frac{1}{2}i\kappa'c\right)^2\right).
\end{aligned}$$

Luego,

$$\begin{aligned}
A_m^{(3)} &= A_m^n - i\kappa'cA_m^{(2)} \\
&= A_m^n - i\kappa'c\left(1 - \frac{1}{2}i\kappa'c + \left(\frac{1}{2}i\kappa'c\right)^2\right)A_m^n \\
&= A_m^n\left(1 - i\kappa'c + \frac{1}{2}(i\kappa'c)^2 - \frac{1}{4}(i\kappa'c)^3\right).
\end{aligned}$$

Ahora haciendo  $z = -i\kappa'c$ .

$$\begin{aligned}
A_m^{(n+1)} &= A_m^n - \frac{\Delta t}{6}\left[G(A_m^n) + 2G(A_m^{(1)}) + 2G(A_m^{(2)}) + G(A_m^{(3)})\right], \\
&= A_m^n - \frac{i\kappa'c}{6}\left[A_m^n + 2A_m^{(1)} + 2A_m^{(2)} + A_m^{(3)}\right] \\
&= A_m^n - \frac{i\kappa'c}{6}\left[A_m^n + 2\left(1 - \frac{i\kappa'c}{2}\right)A_m^n + 2A_m^n\left(1 - \frac{1}{2}i\kappa'c + \left(\frac{1}{2}i\kappa'c\right)^2\right) \right. \\
&\quad \left. + A_m^n\left(1 - i\kappa'c + \frac{1}{2}(i\kappa'c)^2 - \frac{1}{4}(i\kappa'c)^3\right)\right] \\
&= A_m^n\left[1 + \frac{z}{6}\left[1 + 2\left(1 + \frac{z}{2}\right) + 2\left(1 + \frac{z}{2} + \left(\frac{1}{2}z\right)^2\right) \right. \right. \\
&\quad \left. \left. + \left(1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3\right)\right]\right] \\
&= A_m^n\left[1 + \frac{z}{6} + \frac{z}{3}\left(1 + \frac{z}{2}\right) + \frac{z}{3}\left(1 + \frac{z}{2} + \frac{z^2}{4}\right) + \frac{z}{6}\left(1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3\right)\right]
\end{aligned}$$

$$\begin{aligned}
&= A_m^n \left[ 1 + \frac{z}{6} + \frac{z}{3} + \frac{z^2}{6} + \frac{z}{3} + \frac{z^2}{6} + \frac{z^3}{12} + \frac{z}{6} + \frac{z^2}{6} + \frac{z^3}{12} + \frac{z^4}{24} \right] \\
&= A_m^n \left[ 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \right].
\end{aligned}$$

Entonces, el factor de amplificación es

$$G_m = \frac{A_m^{n+1}}{A_m^n} = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}.$$

### 5.4.3. Consideraciones del factor de amplificación

El esquema es estable si

$$|G_m| \leq 1, \quad \forall \kappa \in (0; \pi].$$

Como vimos en la sección 5.2.2 es difícil derivar la condición de estabilidad analítica para los métodos de VF por la complejidad de  $z$  o  $\kappa'$ . Para las Figuras 5.4, y 5.5 los esquemas de reconstrucción propuestos son estables si  $c \leq 1,1$ . De otro lado, cuando  $c \geq 1,2$ , usando RK3 los esquemas de segundo orden y cuarto orden con  $(w_1 = 1, w_2 = 0)$  resultan ser inestables pues se ven oscilaciones muy grandes para números de onda que satisfacen  $|G_m| > 1$ . Para el mismo esquema, con RK4 se controla un poco mejor la inestabilidad. También observamos que el comportamiento inestable cuando usamos un RK3 con números CFL entre 1.1 a 1.2, se traslada para valores entre 1.2 y 1.4 con RK4. Por último, cuando usamos RK3 y RK4 se observa que el esquema de cuarto orden con parámetros  $w_1 = 0,01, w_2 = 0,01$  se comporta mejor que con  $(w_1 = 1, w_2 = 0)$ .

En el siguiente capítulo, nos abocaremos a mostrar a los cálculos computacionales para la ecuación de advección unidimensional donde usaremos los parámetros descritos arriba asegurando la estabilidad de la solución.

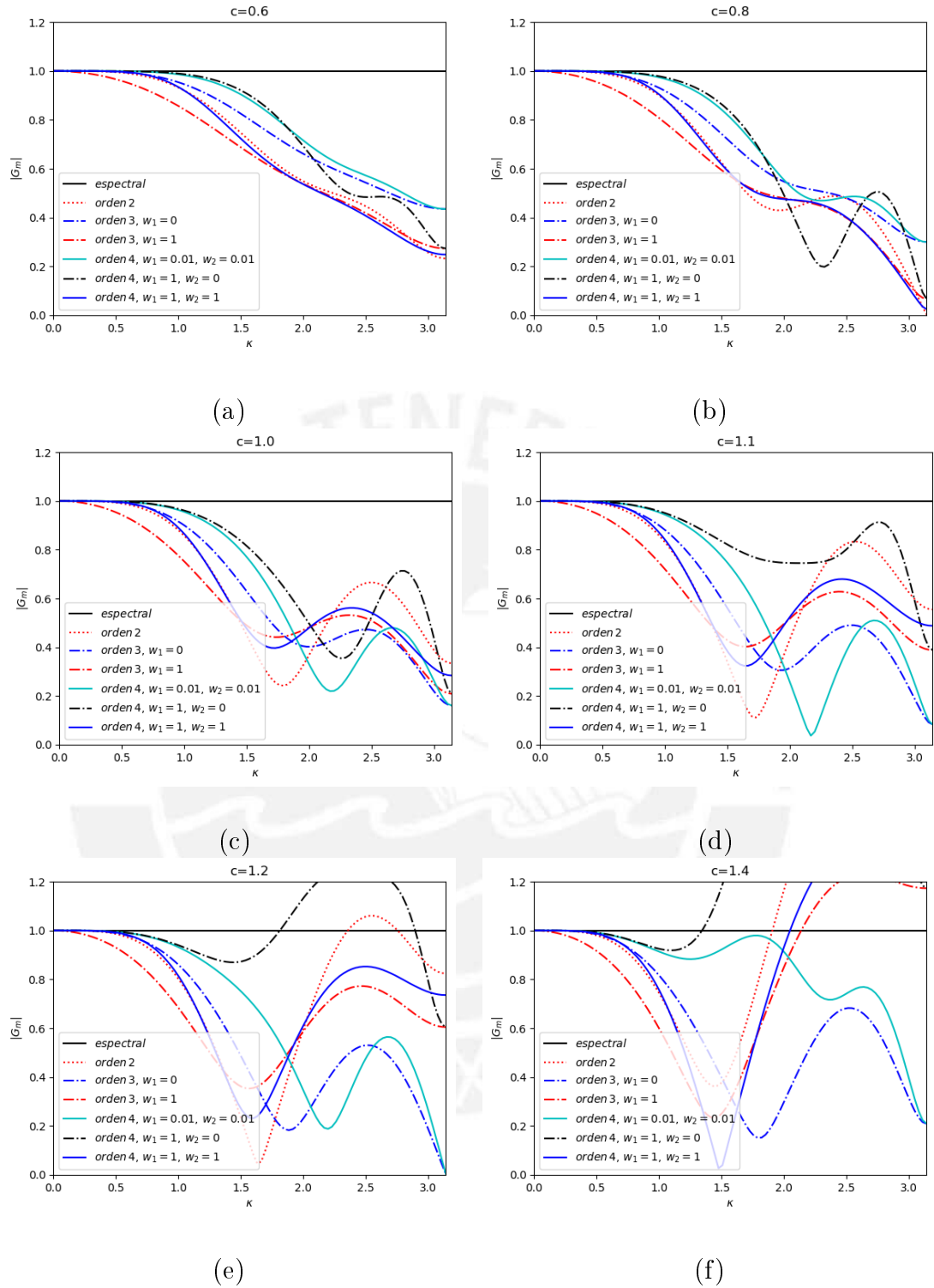


Figura 5.4: Factor de amplificación para los esquemas CLSFV usando TVD RK3. Los números CLF son 0.6, 0.8, 1.0, 1.1, 1.2 y 1.4 en (a) -(f) respectivamente.

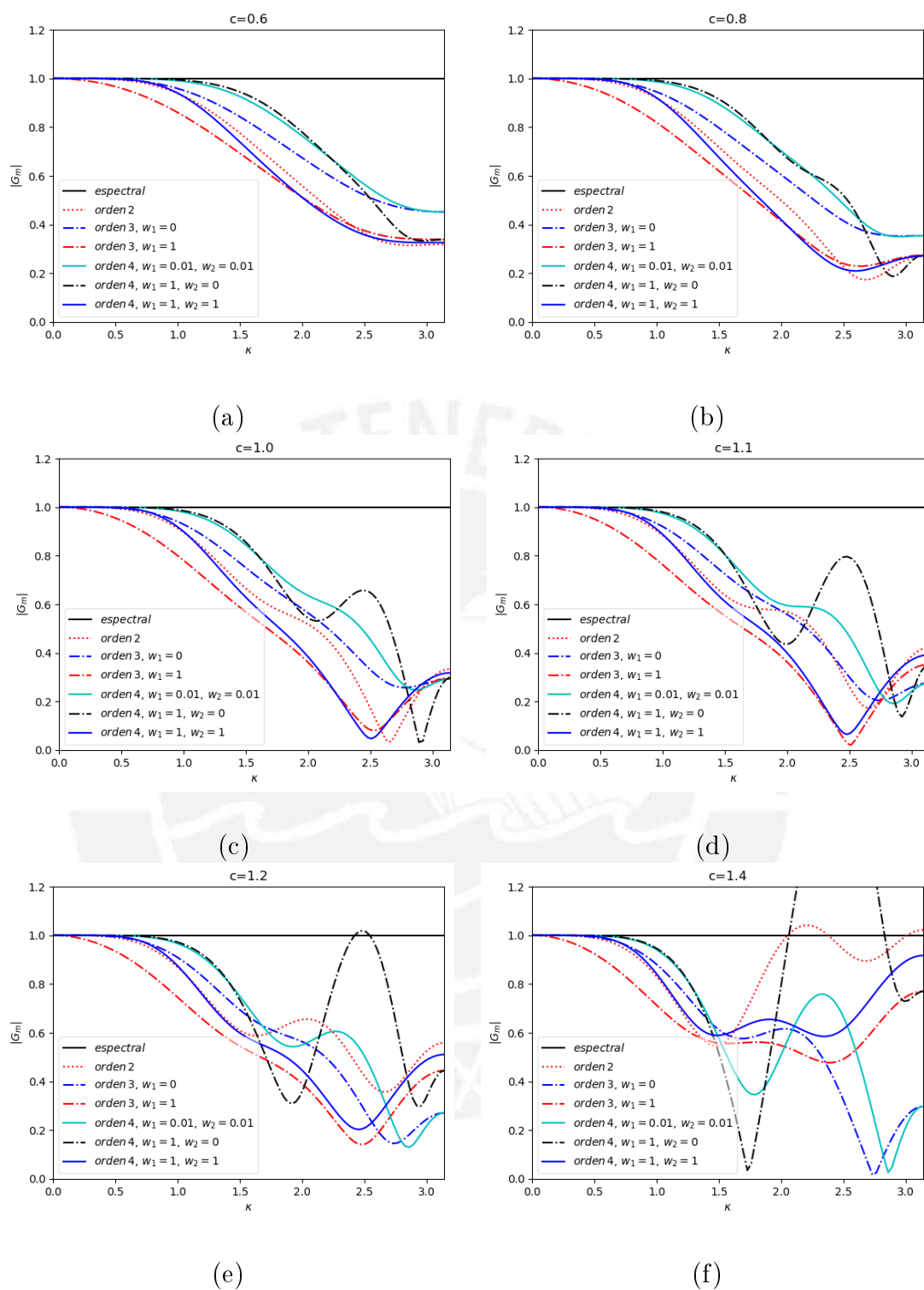


Figura 5.5: Factor de amplificación para los esquemas CLSFV usando RK4. Los números CFL son 0.6, 0.8, 1.0, 1.1, 1.2 y 1.4 en (a) -(f) respectivamente.

## Capítulo 6

### Validación del método

En este capítulo mostramos los resultados numéricos del método de volúmenes finito aplicado a la ecuación diferencial de advección unidimensional

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \\ u(x, 0) = u_0(x) = \text{sen}(2\pi x). \end{cases} \quad (6.1)$$

cuya solución exacta es

$$u(x, t) = u_0(x - t) = \text{sen}(2\pi(x - t)).$$

Como

$$\text{sen}(2\pi x) = \frac{e^{i2\pi x} - e^{-i2\pi x}}{2i} = \frac{1}{2i}e^{i2\pi x} - \frac{1}{2i}e^{-i2\pi x}$$

es combinación lineal de  $e^{ik_1x}$  y  $e^{-ik_1x}$  donde  $k_1 = 2\pi$ , tenemos que la función inicial  $u_0(x)$  tiene un único número de onda. El número de onda escalado  $\kappa = k_1h$  irá disminuyendo conforme disminuyamos el tamaño de la partición.

Dos tipos de condiciones de frontera serán considerados en los dos extremos de la región  $0 \leq x \leq 1$ : una condición de frontera de Dirichlet y la otra periódica.

La evolución en tiempo será hasta cuando  $t = 1$ ; es decir, la solución esperada en el tiempo  $t = 1$  será  $u(x, 1) = \text{sen}(2\pi x)$ . Conforme al análisis de estabilidad visto en la sección 5.4, los esquemas deberán mantenerse estables para el número  $CFL = 1,0$ ; valor elegido para los cálculos de este capítulo.

El error absoluto será calculado en base al Capítulo 3 de Strikwerda, J [18] y estará dado en la siguiente expresión:

$$|error| = \|u - \tilde{u}\|_2 = \sqrt{\frac{\sum_{i=1}^n (u(x_i, 1) - \tilde{u}(x_i, 1))^2}{n}},$$

donde  $\tilde{u}$  es la solución aproximada.

## 6.1. Requerimientos de software y hardware

La implementación del método de volúmenes finitos fue desarrollado en código Python (<http://www.python.org>) en la versión 3.5.1 para 64bits, usando el entorno Spyder(The Scientific PYthon Development EnviRomment) en su versión 2.3.8. Para la programación simbólica se utilizó librería SymPy (<http://www.sympy.org/>). Para mayor información se sugiere revisar el libro de K Jarrod Millman y Michael Aivazis [9].

Los mayores requerimientos de hardware, necesarios para la ejecución de los programas implementados, se usaron para validar las matrices de reconstrucción y el cálculo de los número de ondas modificados estudiados en la sección 3.1 y 5.2.2, respectivamente. Al usarse programación simbólica el tiempo en la ejecución de los programas se elevó aún más. Así, para órdenes bajos se usó una laptop personal con un procesador Intel(R) Core(TM), 8GB de Memoria y sistema operativo Windows 10. de 64 bits. Mientras que para la ejecución de los programas implementados para órdenes intermedios a altos se usó el sistema de súper cómputo denominado Legión, desarrollado por la Dirección de Tecnología de Información de la PUCP .

## 6.2. Implementación y Algoritmo

El algoritmo 1 muestra los pasos generales que sigue el esquema de volúmenes finitos de alto orden. El procedimiento `solveredpdir` (líneas 13 a 22) y el procedimiento `solveredpper` (líneas 23 a 322) son los que iteran los pasos temporales con condiciones de frontera de Dirichlet y periódicas, respectivamente. Para ello primero, debemos calcular los coeficientes de los polinomios de reconstrucción  $u_i^l$



(procedimientos `coefreconstruclidichlet` y `coefreconstrucperiod` líneas 14 y 24 respectivamente) dada por la ecuación (3.2). En las líneas 19, 25 y 29, el procedimiento `evalreconstrucfuncinti` evalúa el polinomio de reconstrucción en el intervalo  $i$ -ésimo. Luego, calcula los flujos numéricos en cada intervalo del dominio computacional (incluidos en parte de la línea 21 y 31) usando la ecuación (4.9) en cada paso del método de Runge-Kutta estudiado en la sección 2.3. En el algoritmo 1 únicamente mostramos a modo de ejemplo la primera etapa de los métodos RK 3. Se procede de manera similar para las otras etapas.

Debemos notar que, para las iteraciones temporales con las condiciones de frontera de Dirichlet, los pasos intermedios en los métodos de Runge-Kutta vistos en la Sección 2.3, específicamente del flujo numérico en la frontera de Dirichlet,  $u_{\frac{1}{2}}^L$ , son mostrados en el Algoritmo 2.

### 6.3. Resultados numéricos

En esta sección presentamos los resultados numéricos obtenidos al iterar los métodos de volúmenes finitos aplicados a la ecuación de advección lineal con los componentes estudiados en los Capítulos 3 y 4. Usamos tanto las condiciones de frontera de Dirichlet como las periódicas estudiadas en las Sección 3.1.1 y 3.1.2, respectivamente. Para cada tipo de condición de frontera se ha considerado en la evolución del tiempo dos tipos de esquemas, el método de Runge-Kutta de tres pasos de orden 3 y el método de Runge-Kutta clásico de cuatro pasos y orden 4 vistos en la Sección 2.3.

En la sección 5.3, se establecieron los parámetros óptimos para mejorar las propiedades de disipación y dispersión en los esquemas de reconstrucción. Estos parámetros óptimos fueron  $w_1 = 0$  en esquemas de tercer orden ( $k = 2$ ), y  $w_1 = 0,01 = w_2$  y  $w_1 = 1, w_2 = 0$  en esquemas de cuarto orden ( $k = 3$ ). También con éstos se consiguieron mejores propiedades de estabilidad de Von Neumann para los métodos de Runge-Kutta de tres y cuatro pasos. Para los demás órdenes y

**input :**  $N, \bar{u}_i, \mathbf{x}_i, \bar{\mathbf{x}}, \mathbf{k}, \mathbf{M}$  y RK : son tamaño de partición, valores promedios, extremos de intervalos, puntos medios de los intervalos, orden del polinomio de reconstrucción, parámetro del método ( siempre considero  $M = k - 1$ ) y Métodos de Runge-Kutta respectivamente.

**output:** Solución aproximada  $\tilde{u}$

```

1  $dx \leftarrow \frac{b-a}{N}, x \in [a, b] = [0, 1];$ 
2  $cfl \leftarrow 1;$ 
3  $dt \leftarrow cfl * dx;$ 
4 /* Cálculo los valores promedio */
5  $\bar{u}_i^0 \leftarrow \frac{1}{h} \int_{I_i} u_0(x) dx;$ 
6 if  $condfr=1$  then
7    $\tilde{u} \leftarrow \text{solverdmdir}(\bar{u}_i^0, \mathbf{x}, \bar{\mathbf{x}}, k, M, N, cfl, RK)$ 
8 end
9 if  $condfr=2$  then
10   $\tilde{u} \leftarrow \text{solverdpper}(\bar{u}_i^0, \mathbf{x}, \bar{\mathbf{x}}, k, M, N, cfl, RK)$ 
11 end
12 /* Funciones solver para condición de frontera de Dirichlet
    y periódica */
13 Function  $\text{solverdmdir}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, k, M, N, cfl, RK)$ 
14    $u_i^l \leftarrow \text{cofreconstrucdirichlet}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, k, M, N);$ 
15    $\tilde{u}_{i-\frac{1}{2}}^L \leftarrow u(0, (j) * dt);$ 
16   /* Etapa 1 */
17   for  $i = 0, N - 1$  do
18      $\hat{f}_{i-\frac{1}{2}} \leftarrow f(\tilde{u}_{i-\frac{1}{2}}^L)$ 
19      $\tilde{u}_{i-\frac{1}{2}}^L \leftarrow \text{evalreconstrucfuncinti}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, u_i^l, k, \mathbf{x}_{i+1}, i);$ 
20      $\hat{f}_{i+\frac{1}{2}} \leftarrow f(\tilde{u}_{i-\frac{1}{2}}^L);$ 
21      $\bar{u}_i \leftarrow \bar{u}_i - cfl * (\hat{f}_{i-\frac{1}{2}} - \hat{f}_{i+\frac{1}{2}});$ 
22   end
23 Function  $\text{solverdpper}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, k, M, N, cfl, RK)$ 
24    $u_i^l \leftarrow \text{cofreconstrucperiod}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, k, M, N);$ 
25    $\tilde{u}_{i-\frac{1}{2}}^L \leftarrow \text{evalreconstrucfuncinti}(\bar{u}_{N-1}, \mathbf{x}, \bar{\mathbf{x}}, u_i^l, k, \mathbf{x}_N, N - 1);$ 
26   /* Etapa 1 */
27   for  $i = 0, N - 1$  do
28      $\hat{f}_{i-\frac{1}{2}} \leftarrow f(\tilde{u}_{i-\frac{1}{2}}^L)$ 
29      $\tilde{u}_{i-\frac{1}{2}}^L \leftarrow \text{evalreconstrucfuncinti}(\bar{u}_i, \mathbf{x}, \bar{\mathbf{x}}, u_i^l, k, \mathbf{x}_{i+1}, i);$ 
30      $\hat{f}_{i+\frac{1}{2}} \leftarrow f(\tilde{u}_{i-\frac{1}{2}}^L);$ 
31      $\bar{u}_i \leftarrow \bar{u}_i - cfl * (\hat{f}_{i-\frac{1}{2}} - \hat{f}_{i+\frac{1}{2}});$ 
32   end

```

**Algoritmo 1:** Método de volúmenes finitos de alto orden para solución de la ecuación (6.1)

```

1 /* Runge-Kutta 3 pasos */
2  $u_{\frac{1}{2}}^{L,1} \leftarrow u(0, j dt)$ 
3  $u_{\frac{1}{2}}^{L,2} \leftarrow u(0, (j + 1) dt)$ 
4  $u_{\frac{1}{2}}^{L,n+1} \leftarrow u(0, (j + 0,5) dt)$ 
5
6 /* Runge-Kutta Clásico 4 */
7  $u_{\frac{1}{2}}^{L,1} \leftarrow u(0, j dt)$ 
8  $u_{\frac{1}{2}}^{L,2} \leftarrow u(0, (j + 0,5) dt)$ 
9  $u_{\frac{1}{2}}^{L,3} \leftarrow u(0, (j + 0,5) dt)$ 
10  $u_{\frac{1}{2}}^{L,n+1} \leftarrow u(0, (j + 1) dt)$ 

```

**Algoritmo 2:** Consideraciones para el cálculo del flujo numérico en la frontera de Dirichlet para los métodos de Runge-Kutta RK3 y RK4.  $u_{\frac{1}{2}}^L$  se muestra en la Figura 4.4.

demás métodos de Runge-Kutta los valores de los parámetros fueron considerados unitarios; es decir,  $w_i = 1$ .

### 6.3.1. Condiciones de frontera de Dirichlet.

En la Figura 6.1 mostramos las aproximaciones con reconstrucciones lineales para diferentes tamaños de malla usando esquemas de Runge-Kutta clásico de orden cuatro para la evolución temporal. Claramente, el refinamiento de la malla muestra soluciones más próximas.

En el Cuadro 6.1, mostramos el error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet para orden 4 ( $k = 3$ ) con parámetros no optimizados  $w_1 = 1 = w_2$ , y optimizados  $w_1 = 0,01$ ,  $w_2 = 0,01$ ;  $w_1 = 1, w_2 \approx 0$ ; usando un esquema de Runge-Kutta de 3 pasos y de orden 3, y  $CFL = 1$ . Como vemos el orden alcanzado para cualquier valor de los parámetros es 3, a pesar de tener un esquema de reconstrucción de cuarto orden. Esto se justifica porque el método de RK únicamente es de tercer orden. En el Cuadro 6.2, se observa cómo se alcanzan los órdenes para esquemas de reconstrucción menores a o iguales a 3.

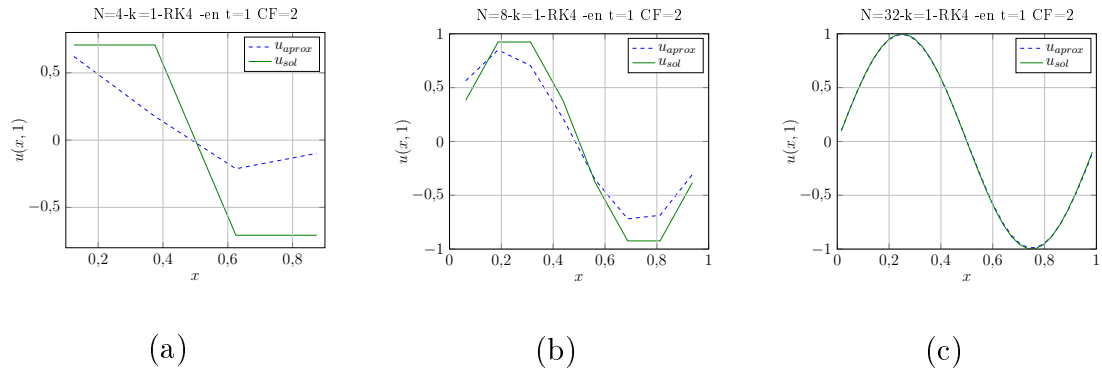


Figura 6.1: Solución de la ecuación advección (2.2) con condiciones de frontera Dirichlet y condición inicial  $u_0(x) = \sin(2\pi x)$ . La solución exacta es  $u(x, t) = u_0(x - t)$  en  $[-1, 1]$  y se muestra las aproximaciones con reconstrucciones lineales para  $N = 4$ (a)  $N = 8$  (b) y  $N = 32$ (c) usando esquemas de Runge-Kutta clásico de orden cuatro para la evolución temporal.

Sin embargo, como se muestra en el Cuadro 6.4, cuando usamos el método de RK 4 de cuarto orden, los órdenes calculados coinciden con los órdenes de reconstrucción tal como se remarcó en la Observación 1 del Capítulo 2 .

Tanto en el Cuadro 6.1 como en el Cuadro 6.3 se muestra como para valores pequeños de  $N$  los parámetros óptimos nos dan menores errores. Este resultado es esperado puesto que en la sección de análisis de Fourier se esperaba que para números de onda altos las reconstrucciones con parámetros optimales tuvieran mejores propiedades dispersivas y disipativas. Sin embargo, en este mismo análisis se vio que para números de onda bajos (valores altos de  $N$ ) no existía mayor mejora.

Cuadro 6.1: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet para orden 4 ( $k = 3$ ) con parámetros no optimizados  $w_1 = 1 = w_2$ , y optimizados  $w_1 = 0,01$ ,  $w_2 = 0,01$ ;  $w_1 = 1, w_2 \approx 0$  ; con RK3 y  $CFL = 1$ .

N	$w_1 = 1 = w_2$		$w_1 = 0,01 = w_2$		$w_1 = 1, w_2 \approx 0$	
	error	orden	error	orden	error	orden
4	1.00E+00		1.00E+00		1.00E+00	
8	1.25E-01	3.01	6.07E-02	4.05	4.61E-02	4.44
16	9.74E-03	3.68	7.43E-03	3.03	7.52E-03	2.61
32	1.13E-03	3.11	1.09E-03	2.77	1.09E-03	2.79
64	1.38E-04	3.03	1.39E-04	2.97	1.38E-04	2.97
128	1.71E-05	3.01	1.74E-05	3.00	1.73E-05	3.00
256	2.15E-06	3.00	2.17E-06	3.00	2.16E-06	3.00
512	2.69E-07	3.00	2.71E-07	3.00	2.70E-07	3.00

Cuadro 6.2: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet, RK3 y  $CFL = 1$ .

N	k=0		k=1		k=2		k=3	
	error	orden	error	orden	$w_1 \approx 0$		$w_1 = 0,01; w_2 = 0,01$	
					error	orden	error	orden
4	7.37E-01		7.23E-01		8.48E-01		1.00E+00	
8	5.51E-01	0.42	2.52E-01	1.52	1.29E-01	2.72	6.07E-02	4.05
16	3.79E-01	0.54	3.30E-02	2.93	2.04E-02	2.66	7.43E-03	3.03
32	2.31E-01	0.71	5.09E-03	2.70	2.92E-03	2.80	1.09E-03	2.77
64	1.30E-01	0.83	1.52E-03	1.74	4.13E-04	2.82	1.39E-04	2.97
128	6.95E-02	0.90	4.47E-04	1.77	5.74E-05	2.85	1.74E-05	3.00
256	3.60E-02	0.95	1.21E-04	1.88	3.08E-05	0.90	2.17E-06	3.00
512	1.83E-02	0.97	3.15E-05	1.94	8.34E-06	1.89	2.71E-07	3.00

Cuadro 6.3: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet para tercer ( $k = 2$ ) y cuarto orden ( $k = 3$ ) con parámetros  $w_1 = 0,00001 \approx 0$ ;  $w_1 = 1 = w_2$  y  $w_1 = 0,01 = w_2$ , RK4 y CFL=1

N	k=2				k=3			
	$w_1 = 1$		$w_1 \approx 0$		$w_1 = 1 = w_2$		$w_1 = 0,01 = w_2$	
	error	orden	error	orden	error	orden	error	orden
4	7.57E-01		8.94E-01		1.21E+00		1.21E+00	
8	2.25E-01	1.75	1.26E-01	2.83	1.91E-01	2.66	1.40E-01	3.11
16	4.51E-02	2.32	1.66E-02	2.92	8.01E-03	4.57	3.81E-03	5.19
32	6.81E-03	2.73	2.08E-03	3.00	2.49E-04	5.01	1.41E-04	4.75
64	9.14E-04	2.90	2.66E-04	2.97	1.15E-05	4.44	7.60E-06	4.22
128	1.18E-04	2.96	5.25E-05	2.34	7.64E-07	3.91	5.58E-07	3.77
256	1.49E-05	2.98	1.87E-05	1.49	5.75E-08	3.73	4.66E-08	3.58
512	1.87E-06	2.99	7.45E-06	1.33	4.57E-09	3.65	4.04E-09	3.53

Cuadro 6.4: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera de Dirichlet con RK4 y CFL=1 .

N	k=0		k=1		k=2)		k=3	
					$w_1 = 0,000001$		$w_1 = 0,01 = w_2$	
	error	orden	error	orden	error	orden	error	orden
4	6.38E-01		6.76E-01		8.94E-01		1.21E+00	
8	5.35E-01	0.26	2.14E-01	1.66	1.26E-01	2.83	1.40E-01	3.11
16	3.75E-01	0.51	2.48E-02	3.11	1.66E-02	2.92	3.81E-03	5.19
32	2.31E-01	0.70	4.81E-03	2.36	2.08E-03	3.00	1.41E-04	4.75
64	1.30E-01	0.83	1.57E-03	1.62	2.66E-04	2.97	7.60E-06	4.22
128	6.95E-02	0.90	4.56E-04	1.78	5.25E-05	2.34	5.58E-07	3.77
256	3.60E-02	0.95	1.23E-04	1.90	1.87E-05	1.49	4.66E-08	3.58
512	1.83E-02	0.97	3.17E-05	1.95	7.45E-06	1.33	4.04E-09	3.53

### 6.3.2. Condiciones de frontera periódicas

Usando el método de Runge-Kutta de cuatro pasos tenemos las soluciones aproximadas en las Figuras 6.2a, 6.2b y 6.2c para  $N = 4$ , 8 y 32 respectivamente.

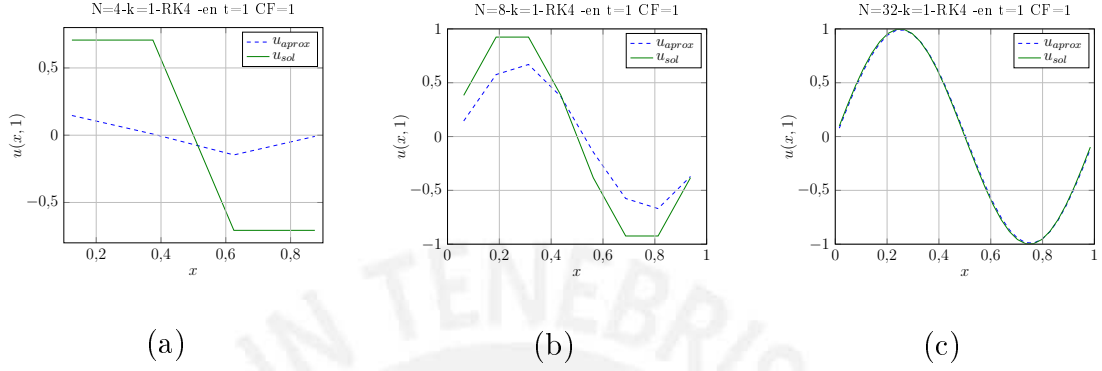


Figura 6.2: Solución de la ecuación advección (2.2) con condiciones de frontera periódicas y condición inicial  $u_0(x) = \sin(2\pi x)$ . La solución exacta es  $u(x, t) = u_0(x - t)$  en  $[-1, 1]$  y se muestra las aproximaciones con reconstrucciones lineales para  $N = 4$ (a)  $N = 8$  (b) y  $N = 32$ (c) usando esquemas de Runge-Kutta clásico de orden cuatro para la evolución temporal.

En los Cuadros 6.5, 6.6, 6.8 y 6.7 mostramos los errores de aproximación y órdenes de convergencia para la ecuación de advección lineal con condiciones de frontera periódicas, para esquemas de reconstrucción de segundo, tercero y cuarto orden ( $k = 3$ ) con parámetros no optimizados  $w_1 = 1, w_2 = 1$  y optimizados  $w_1 = 0,01, w_2 = 0,01$ ;  $w_1 = 1; w_2 = 0$  usando métodos de Runge-Kutta de tercer y cuarto orden. Los resultados son similares a los presentados a la sección anterior.

En el Cuadro 6.5 se usa un esquema de cuarto orden en la reconstrucción con diferentes valores para los parámetros y un método de RK de orden 3 para la evolución temporal. En todos los casos nuevamente se consigue un orden 3 en la convergencia del método de volumen Finito. Ahora como en la sección anterior, tanto en el Cuadro 6.5 como en el Cuadro 6.6 los órdenes de convergencia para esquemas de reconstrucción menores a tres son alcanzados. En estos mismo cuadros también se observa que para los parámetros optimales los errores de aproximación son menores cuando se tiene mallas gruesas.

Cuadro 6.5: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos, para esquemas de reconstrucción de cuarto orden ( $k = 3$ ) con parámetros no optimizados  $w_1 = 1, w_2 = 1$  y optimizados  $w_1 = 0,01, w_2 = 0,01$ ;  $w_1 = 1; w_2 = 0$ ; con RK3 y CFL=1

N	$w_1 = 1 = w_2$		$w_1 = 0,01 = w_2$		$w_1 = 1, w_2 = 1.0E-5$	
	error	orden	error	orden	error	orden
4	7.22E-01		5.47E-01		6.89E-01	
8	2.20E-01	1.72	1.02E-01	2.42	1.03E-01	2.74
16	1.86E-02	3.56	1.19E-02	3.10	1.18E-02	3.13
32	1.67E-03	3.48	1.42E-03	3.07	1.41E-03	3.06
64	1.83E-04	3.19	1.76E-04	3.02	1.75E-04	3.01
128	2.20E-05	3.05	2.19E-05	3.00	2.19E-05	3.00
256	2.74E-06	3.01	2.74E-06	3.00	2.73E-06	3.00
512	3.42E-07	3.00	3.42E-07	3.00	3.42E-07	3.00

En el Cuadro 6.7, gracias a mejoras en la programación conseguimos tener mallas muy refinadas. En este cuadro se muestra los errores de aproximación y el orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción orden cuarto ( $k = 3$ ) con parámetros no óptimos  $w_1 = 1 = w_2$  y óptimos  $w_1 = 0,01 = w_2$ , usando el método de Runge-Kutta de cuatro pasos clásico y  $CFL = 1$ . Tanto en este cuadro y, más evidente, en la Figura 6.3 se muestra una reducción en el error cuando se usan los parámetros óptimos. También se observa en esta figura que las líneas de tendencia de los errores son diferentes hasta un tamaño de malla con  $N = 256$ , para luego mantener su tendencia al orden de convergencia esperado.

En el Cuadro 6.8, se muestran los errores de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción con diferentes órdenes, con parámetros óptimos cuando  $k = 2, 3$  y  $w_i = 1, 0$  para  $k = 5, 6$ ; usando un esquema RK4 y CFL=1. En este cuadro, como en la Figura 6.4, ocurren comportamientos similares cuando



Cuadro 6.6: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción de diferentes órdenes con RK3 y CFL=1

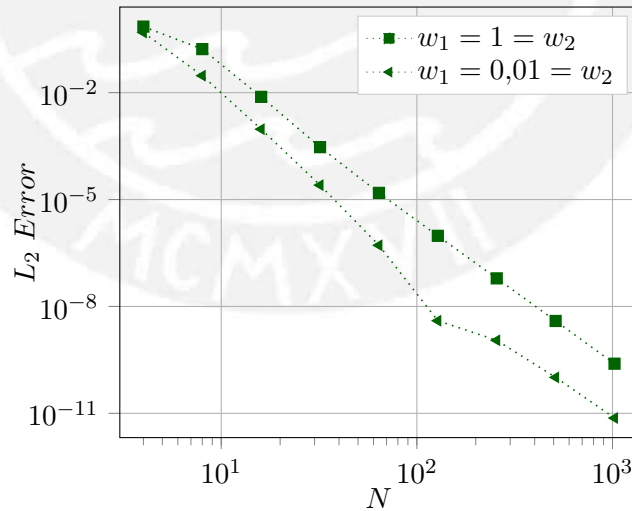
N	k=0		k=1		k=2 ( $w_1 = 1,0E - 6$ )		k=3( $w_1 = 0,01 = w_2$ )	
	error	orden	error	orden	error	orden	error	orden
4	7.30E-01		6.97E-01		6.78E-01		5.47E-01	
8	6.84E-01	0.09	2.81E-01	1.31	2.11E-01	1.68	1.02E-01	2.42
16	5.21E-01	0.39	5.97E-02	2.24	3.25E-02	2.70	1.19E-02	3.10
32	3.34E-01	0.64	1.39E-02	2.11	4.21E-03	2.95	1.42E-03	3.07
64	1.91E-01	0.80	3.48E-03	1.99	5.29E-04	2.99	1.76E-04	3.02
128	1.03E-01	0.90	8.83E-04	1.98	6.62E-05	3.00	2.19E-05	3.00
256	5.32E-02	0.95	2.23E-04	1.99	8.26E-06	3.00	2.74E-06	3.00
512	2.71E-02	0.97	5.61E-05	1.99	1.03E-06	3.00	3.42E-07	3.00

usamos condiciones de frontera de Dirichlet, en el sentido que el orden alcanzado es el menor entre el orden de reconstrucción y el del esquema RK (Ver Observación 1 del Capítulo 2). Basta con observar el orden de convergencia cuando usamos órdenes de reconstrucción quinto y sexto; en estos casos el orden calculado a lo más coincide con el orden del método de Runge-Kutta; es decir, orden cuatro.

Cuadro 6.7: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción orden cuarto ( $k = 3$ ) con parámetros no óptimos  $w_1 = 1 = w_2$  y óptimos  $w_1 = 0,01 = w_2$ , RK4 y CFL=1

N	$w_1 = 1 = w_2$		$w_1 = 0,01 = w_2$	
	error	orden	error	orden
4	7.15E-01		4.89E-01	
8	1.67E-01	2.09	3.00E-02	4.03
16	7.63E-03	4.46	9.45E-04	4.99
32	2.96E-04	4.69	2.54E-05	5.22
64	1.54E-05	4.27	5.20E-07	5.61
128	9.59E-07	4.01	3.97E-09	7.03
256	6.14E-08	3.97	1.12E-09	1.82
512	3.91E-09	3.97	1.02E-10	3.46
1024	2.47E-10	3.99	7.37E-12	3.79

Figura 6.3: Error de aproximación para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción orden cuarto ( $k = 3$ ) con parámetros no óptimos  $w_1 = 1 = w_2$  y óptimos  $w_1 = 0,01 = w_2$ , RK4 y CFL=1



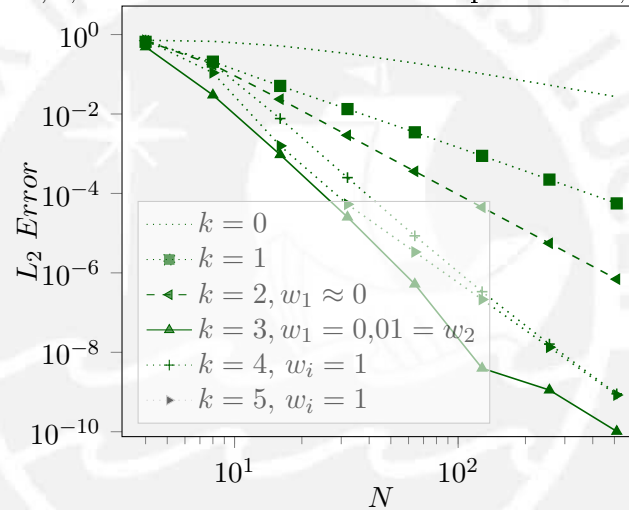
Cuadro 6.8: Error de aproximación y orden de convergencia para la ecuación de advección lineal con condiciones de frontera periódicos para esquemas de reconstrucción con diferentes órdenes, con parámetros óptimos cuando  $k = 2, 3$  y  $w_i = 1, 0$  para  $k = 5, 6$ ; usando RK4 y CFL=1

N	k=0		k=1		k=2 ( $w_1 = 1.0E-5$ )	
	error	orden	error	orden	error	orden
4	7.18E-01		6.57E-01		6.99E-01	
8	6.74E-01	0.09	2.08E-01	1.66	1.75E-01	2.00
16	5.18E-01	0.38	5.09E-02	2.03	2.36E-02	2.89
32	3.33E-01	0.64	1.33E-02	1.93	2.91E-03	3.02
64	1.91E-01	0.80	3.46E-03	1.95	3.60E-04	3.02
128	1.03E-01	0.90	8.83E-04	1.97	4.47E-05	3.01
256	5.32E-02	0.95	2.23E-04	1.98	5.56E-06	3.00
512	2.71E-02	0.97	5.61E-05	1.99	6.94E-07	3.00

N	k=3 ( $w_1 = 0,01, w_2 = 0,01$ )		k=4 ( $w_i = 1$ )		k=5 ( $w_i = 1$ )	
	error	orden	error	orden	error	orden
4	4.89E-01		7.26E-01		7.22E-01	
8	3.00E-02	4.03	1.85E-01	1.97	1.09E-01	2.73
16	9.45E-04	4.99	7.68E-03	4.59	1.58E-03	6.11
32	2.53E-05	5.22	2.50E-04	4.94	5.32E-05	4.89
64	5.20E-07	5.61	8.55E-06	4.87	3.36E-06	3.98
128	3.97E-09	7.03	3.36E-07	4.67	2.14E-07	3.98
256	1.12E-09	1.82	1.61E-08	4.38	1.34E-08	3.99
512	1.02E-10	3.46	9.00E-10	4.16	8.40E-10	4.00

Figura 6.4: Orden de convergencia para solución de la ecuación de advección uni-dimensional para diferentes órdenes con parámetros óptimos cuando  $k = 2, 3$ ;  $w_i = 1, 0$  para  $k = 5, 6$ ; con condiciones de frontera periódicas, CFL=1 y RK4.



## Capítulo 7

### Conclusiones y trabajos futuros

En esta tesis se estudiaron los esquemas de volúmenes finitos compactos de alto orden para el caso unidimensional de una ecuación de advección y se implementó un código en lenguaje de programación Python. Para estudiar los esquemas de volúmenes finitos establecimos la forma integral de la ecuación de advección lineal unidimensional, lo que nos permitió dar las condiciones necesarias para obtener un esquema de reconstrucción de alto orden, comprobando además el orden de convergencia.

Estudiamos los problemas de Riemann para la ecuación de advección unidimensional permitiéndonos justificar el cálculo del flujo numérico. Con ello pudimos evolucionar en el tiempo usando esquemas de Runge-Kutta de órdenes adecuados teniendo presente las condiciones de frontera y periodicidad.

El análisis espectral fue un elemento importante para entender la necesidad de plantear esquemas numéricos, de modo que los números de onda se aproximen a los exactos. Junto con un proceso de optimización, se encontraron los parámetros adecuados en los esquemas de reconstrucción. Esta parte del trabajo desarrollado fue la que más demandó tiempo, tanto en la implementación, como en el tiempo de ejecución del programa. Al realizar cálculos simbólicos sobre matrices de reconstrucción, que incluían cálculo de inversas de matrices, se obtuvieron los números de onda modificados analíticos presentados en la sección 5.2.2. Aquí fue oportuna la colaboración del grupo Legión, que administran un cluster de la PUCP y pudimos obtener estos números en esquemas de alto orden. Desafortunadamente, no

se consiguió resultados para órdenes más altos que los presentados en el artículo de Wang [22] por la complejidad de los cálculos.

De otro lado la validación de los órdenes de convergencia de la solución numérica de la ecuación de advección unidimensional fue conseguida, puesto que siempre coincidía con el menor entre el orden de la reconstrucción y el del esquema de RK como era esperado. Afortunadamente conseguimos acelerar los cálculos al explicitar las matrices de reconstrucción para mallas uniformes. Así el tiempo de ejecución se redujo considerablemente e incluso conseguimos validar el orden de convergencia teniendo números de particiones del orden  $2^{10}$  sin recurrir al Cluster de la universidad.

También es importante indicar que hemos hecho un aporte significativo al trabajo hecho por Wang puesto que él únicamente estudió soluciones con condiciones de frontera periódica y en nuestro caso añadimos el estudio a las condiciones de frontera de Dirichlet.

## Trabajos futuros

Gracias a explicitar las matrices de reconstrucción en los programas desarrollados acortamos los tiempo de ejecución, con lo que estamos en condiciones de hacer un estudio comparativo de los tiempos de ejecución para cada esquema de reconstrucción.

Wang muestra los números de onda analíticamente gracias al cálculo simbólico y creemos que adaptando mejor el programa a una programación paralela, podemos conseguir resultados para órdenes mayores usando apropiadamente el Cluster de la PUCP.

Luego de estudiar la ecuación de advección unidimensional podemos incluir más adelante la parte difusiva en las ecuaciones que provienen de las leyes de conservación visto en la sección 2.1 y posteriormete estudiar la ecuación de Euler de la mecánica de fluidos.

Así mismo, resultaría conveniente estudiar maneras de acelerar los procesos iterativos. Aquí los métodos p-multigrad, de principios similares a los métodos multigrad estudiados en [26], pueden resultar adecuados conforme lo indica Casavilca [3].

# Apéndice A

## Anexo

En este anexo incluimos definiciones básicas tomadas del libro de E. Toro [19] necesarias para entender el cálculo del flujo numérico, componente importante en los métodos de volúmenes finitos.

### A.1. Definiciones

**Definición 4.** (Leyes de conservación) Las leyes de conservación son un sistema de ecuaciones diferenciales parciales que pueden ser escritas en la forma

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0 \quad (\text{A.1})$$

donde

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}. \quad (\text{A.2})$$

$\mathbf{U}$  es llamado vector de variables conservativas,  $\mathbf{F}(\mathbf{U})$  es el vector de flujos y cada una de sus componentes  $f_i$  es una función de las componentes  $u_j$  de  $\mathbf{U}$ .

**Definición 5. (Matriz Jacobiana)** El Jacobiano de una función flujo  $\mathbf{F}(\mathbf{U})$  en (A.2) es la matriz

$$\mathbf{A}(\mathbf{U}) = \partial \mathbf{F} / \partial \mathbf{U} = \begin{bmatrix} \partial f_1 / \partial u_1 & \cdots & \partial f_1 / \partial u_m \\ \partial f_2 / \partial u_1 & \cdots & \partial f_2 / \partial u_m \\ \vdots & \vdots & \vdots \\ \partial f_m / \partial u_1 & \cdots & \partial f_m / \partial u_m \end{bmatrix}. \quad (\text{A.3})$$

Las entradas  $a_{i,j}$  de  $\mathbf{A}(\mathbf{U})$  son las derivadas parciales de las componentes  $f_i$  de los vectores  $\mathbf{F}$  con respecto a las componentes  $u_j$  de los vectores de las variables conservativas  $\mathbf{U}$ ; es decir,  $a_{i,j} = \partial f_i / \partial u_j$ .

**Definición 6. (Valores propios)** Los valores propios  $\lambda_i$  de una matriz  $\mathbf{A}$  son la solución del polinomio característico

$$|\mathbf{A} - \lambda \mathbf{I}| = \det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

donde  $\mathbf{I}$  es la matriz identidad. Los valores propios de la matriz de coeficientes  $\mathbf{A}$  de un sistema de la forma (A.1) son llamados valores propios del sistema.

**Definición 7. (Vectores propios)** Un vector propio por derecha de la matriz  $\mathbf{A}$  correspondiente al valor propio  $\lambda$  de  $\mathbf{A}$  es un vector  $\mathbf{K} = [k_1, k_2, \dots, k_m]^T$  que satisface

$$\mathbf{A} \mathbf{K} = \lambda \mathbf{K}.$$

Similarmente, un vector propio por izquierda de la matriz  $\mathbf{A}$  correspondiente al valor propio  $\lambda$  de  $\mathbf{A}$  es un vector  $\mathbf{L} = [l_1, l_2, \dots, l_m]^T$  que satisface

$$\mathbf{L} \mathbf{A} = \lambda \mathbf{L}.$$

**Definición 8. (Sistemas Hiperbólicos)** Un sistema de la forma (A.1) se dice que es hiperbólico en cada punto  $(x; t)$  si  $\mathbf{A}$  tiene  $m$  valores propios reales  $\lambda_1, \lambda_2, \dots, \lambda_m$  y un conjunto correspondiente de  $m$  vectores propios linealmente independientes  $\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \dots, \mathbf{K}^{(m)}$ . El sistema se dice que es estrictamente hiperbólico si los valores propios  $\lambda_i$  son todos distintos.



**Definición 9. (Sistema diagonalizable)** Una matriz  $\mathbf{A}$  es diagonalizable si puede ser expresada como

$$\mathbf{A} = \mathbf{K}\mathbf{\Lambda}\mathbf{K}^{-1} \text{ o } \mathbf{\Lambda} = \mathbf{K}^{-1}\mathbf{A}\mathbf{K},$$

en términos de una matriz diagonal  $\mathbf{\Lambda}$  y una matriz  $\mathbf{K}$ . Los elementos de la diagonal de la  $\mathbf{\Lambda}$  son los valores propios de la matriz  $\mathbf{A}$  y las columnas  $\mathbf{K}^{(i)}$  de  $\mathbf{K}$  son los vectores propios por derecha de  $\mathbf{A}$  correspondientes a los valores propios  $\lambda_i$ ; es decir,

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} \quad \mathbf{K} = [\mathbf{K}^{(1)}, \dots, \mathbf{K}^{(m)}], \quad \mathbf{A}\mathbf{K}^{(i)} = \lambda_i\mathbf{K}^{(i)}.$$

## A.2. El Problema de Riemman

El problema de Riemman y su solución exacta es una referencia valiosa y útil para evaluar el rendimiento de métodos numéricos y verificar la efectividad de los programas [19]. En esta sección, estudiaremos la ecuación de Euler unidimensional y expondremos los elementos más importantes del problema de Riemman de un sistema de ecuaciones hiperbólicos.

### A.2.1. Ecuación de Euler unidimensional

Denotemos a las variables primitivas o físicas

- $\rho(x, t)$  : densidad o densidad de masa;
- $p(x, t)$ : presión;
- $u(x, t)$ : velocidad en dirección  $x$ .

De modo que podemos proporcionar las así llamadas variable de conservación, la densidad de masa  $\rho$ , momento  $\rho u$  y la Energía total por unidad de masa  $E$ . Físicamente, estas cantidades resultan de la aplicación de las leyes fundamentales

de la conservación de la masa, la segunda ley de Newton y la ley de conservación de la energía. Esto da lugar a una gran clase de métodos numéricos llamados métodos conservativos y dentro de ellos el método de volúmenes finitos que estudiamos en esta tesis.

Las tres leyes de conservación que rigen para el caso unidimensional son

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0, \quad (\text{A.4})$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + p) = 0, \quad (\text{A.5})$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}(u(E + p)) = 0; \quad (\text{A.6})$$

que escrito en su forma matricial sería

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{0} \quad (\text{A.7})$$

llamado un sistema de leyes de conservación, donde  $\mathbf{U}$  es el vector de las variables conservativas y  $\mathbf{F}$  es el vector de flujos definidos respectivamente por

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{F} = \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}.$$

donde

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{A}(\mathbf{U}) \frac{\partial \mathbf{U}}{\partial x}$$

con  $\mathbf{A}(\mathbf{U})$  la matriz Jacobiana (Ver definición 5).

Un caso muy particular es la ecuación de advección lineal, algunas veces llamada convección lineal,

$$u_t + a u_x = 0 \quad (\text{A.8})$$

donde  $f(u) = au$  y  $a$  es la velocidad de propagación constante de la onda. Esta ecuación también es conocida como la ecuación de onda unidimensional.

### A.2.2. Sistemas de ecuaciones lineales hiperbólicos

Ahora extenderemos el análisis a un conjunto de  $m$  ecuaciones diferenciales parciales hiperbólicas de la forma

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{0} \quad (\text{A.9})$$

donde la matriz de coeficientes  $\mathbf{A}$  es constante y como estamos asumiendo que es una ecuación de tipo hiperbólico entonces  $\mathbf{A}$  debe tener  $m$  valores propios reales  $\lambda_i$  y  $m$  vectores propios linealmente independientes  $\mathbf{K}^{(i)}$ .

#### Diagonalización y variables características

Para analizar y resolver el problema de valor inicial (A.9) haremos una transformación de la variable dependiente  $\mathbf{U}(x, t)$  a un nuevo conjunto de variables dependientes  $\mathbf{W}(x, t)$ . Decimos que el sistema (A.9) es diagonalizable si la matriz  $\mathbf{A}$  es diagonalizable (ver definición 9); es decir,  $\mathbf{A}$  puede ser expresado por

$$\mathbf{A} = \mathbf{K} \mathbf{\Lambda} \mathbf{K}^{-1}$$

en términos de una matriz diagonal  $\mathbf{\Lambda}$  y una matriz  $\mathbf{K}$ .

La existencia de la matriz inversa  $\mathbf{K}^{-1}$  hace posible definir el nuevo conjunto de variables dependientes  $\mathbf{W} = (w_1, w_2, \dots, w_m)^T$  llamadas variables características, usando la transformación

$$\mathbf{W} = \mathbf{K}^{-1} \mathbf{U} \quad \text{o} \quad \mathbf{U} = \mathbf{K} \mathbf{W}.$$

Así el sistema lineal (A.9) puede ser expresando en términos de  $\mathbf{W}$  y tendríamos un sistema desacoplado. Luego, como

$$\mathbf{U}_t = \mathbf{K} \mathbf{W}_t, \quad \mathbf{U}_x = \mathbf{K} \mathbf{W}_x,$$

reemplazando tenemos

$$\mathbf{K} \mathbf{W}_t + \mathbf{A} \mathbf{K} \mathbf{W}_x = \mathbf{0}.$$

Multiplicando por  $\mathbf{K}^{-1}$  quedaría

$$\mathbf{W}_t + \Lambda \mathbf{W}_x = \mathbf{0}.$$

o equivalentemente

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}_t + \begin{bmatrix} \lambda_1 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}_x = 0$$

lo que nos lleva a tener un sistema de EDP

$$\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0, \quad i = 1, \dots, m.$$

Este sistema de ecuaciones desacoplados son idénticos a la ecuación de advección lineal (4.1), donde la velocidad característica es  $\lambda_i$  y las  $m$  curvas características satisfacen la EDO

$$\frac{dx}{dt} = \lambda_i \quad i = 1, \dots, m$$

y si consideramos un PVI con  $\mathbf{U}^{(0)} = (u_1^{(0)}, \dots, u_m^{(0)})$ , y  $\mathbf{W}^{(0)} = \mathbf{K}^{-1} \mathbf{U}^{(0)} = (w_1^{(0)}, \dots, w_m^{(0)})$  las soluciones serían

$$w_i(x, t) = w_i^{(0)}(x - \lambda_i t), \quad i = 1, \dots, m.$$

de manera que como  $\mathbf{U} = \mathbf{K} \mathbf{W}$ , entonces podemos expandir la solución  $\mathbf{U}$  del siguiente modo

$$\mathbf{U}(x, t) = \sum_{i=1}^m w_i^{(0)}(x - \lambda_i t) \mathbf{K}^{(i)}. \quad (\text{A.10})$$

Así dado un punto  $(x, t)$  en el plano  $x - t$ , la solución  $\mathbf{U}(x, t)$  en ese punto depende sólo de los datos iniciales en los  $m$  puntos  $x_0^{(i)} = x - \lambda_i t$ . Estos son las intersecciones de las características de velocidad  $\lambda_i$  con el eje  $x$ . La solución (A.10) para  $\mathbf{U}$  podemos verla como una superposición de  $m$  ondas, cada una de las cuales

es independientemente advectiva sin cambiar su forma. La  $i$ -ésima onda tiene la forma de  $w_i^{(0)}(x)\mathbf{K}^{(i)}$  y se propaga con una velocidad  $\lambda_i$ .

### A.2.3. El Problema de Riemman para un sistema de ecuaciones hiperbólicos.

El Problema de Riemman para la ecuación hiperbólica con un sistema de coeficientes constantes está dado por el problema de valor inicial

$$\begin{cases} EDP: & \mathbf{U}_t + \mathbf{A} \mathbf{U}_x = 0, \quad x \in \mathbb{R}, t > 0 \\ CI & \mathbf{U}(x, 0) = \mathbf{U}^{(0)}(x) = \begin{cases} \mathbf{U}_L, & \text{si } x < 0, \\ \mathbf{U}_R, & \text{si } x > 0. \end{cases} \end{cases} \quad (\text{A.11})$$

Se asume que el sistema es estrictamente hiperbólico y por tanto, podemos considerar los valores propios como

$$\lambda_1 < \lambda_2 < \cdots < \lambda_m.$$

### Solución general

En la Figura (A.1) se muestra la forma de la solución general del problema de Riemman (A.11) en el plano  $x - t$ . Consiste de  $m$  ondas que salen del origen, uno por cada valor propio  $\lambda_i$ . Cada onda  $i$  lleva un salto de discontinuidad en  $\mathbf{U}$  que se propaga con una velocidad  $\lambda_i$ . La solución de la izquierda de la onda  $\lambda_1$  es el dato inicial  $\mathbf{U}_L$  y el de la derecha de la onda  $\lambda_m$  es  $\mathbf{U}_R$ . El problema yace en determinar la solución en la parte comprendida entre las ondas  $\lambda_1$  y  $\lambda_m$ .

Como los vectores propios  $\mathbf{K}^{(1)}, \dots, \mathbf{K}^{(m)}$  son linealmente independientes, podemos expandir los datos  $\mathbf{U}_L$  y  $\mathbf{U}_R$  como combinación lineal de ellos, es decir

$$\mathbf{U}_L = \sum_{i=1}^m \alpha_i \mathbf{K}^{(i)}, \quad \mathbf{U}_R = \sum_{i=1}^m \beta_i \mathbf{K}^{(i)} \quad (\text{A.12})$$

con coeficiente constantes  $\alpha_i$  y  $\beta_i$  con  $i = 1, \dots, m$ . La solución de la ecuación (A.11) está dado por (A.10) en términos de los datos iniciales  $w_i^{(0)}(x)$  para las

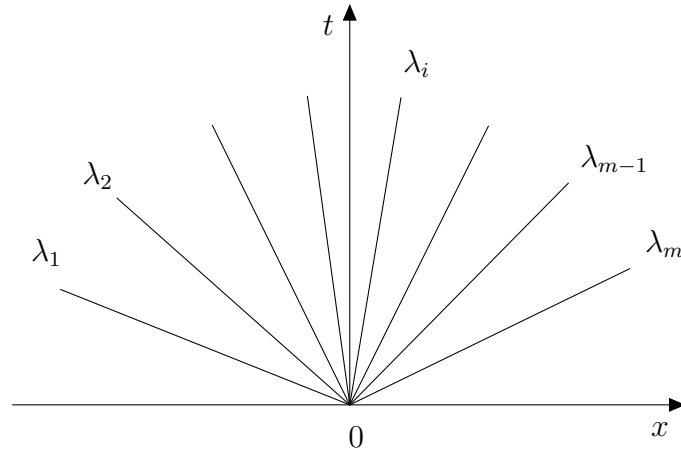


Figura A.1: Estructura de la solución del problema de Riemman para un sistema lineal hiperbólico general  $m \times m$  con coeficientes constantes.

variables características y los vectores propios por derecha  $\mathbf{K}^{(i)}$ . Ahora en términos de las variables característica tenemos  $m$  problemas de Riemman

$$\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0, \quad i = 1, \dots, m$$

con los datos iniciales

$$w_i^{(0)}(x) = \begin{cases} \alpha_i, & \text{si } x < 0, \\ \beta_i, & \text{si } x > 0 \end{cases}, \quad i = 1, \dots, m$$

Luego la solución de la ecuación es

$$w_i(x, t) = w_i^{(0)}(x - \lambda_i t) = \begin{cases} \alpha_i, & \text{si } x - \lambda_i t < 0, \\ \beta_i, & \text{si } x - \lambda_i t > 0 \end{cases}.$$

Para un punto dato  $(x, t)$  existe un valor propio  $\lambda_I$  tal que  $\lambda_I < \frac{x}{t} < \lambda_{I+1}$ , que es  $x - \lambda_i t > 0, \forall i$  con tal  $i < I$ . Así podemos escribir la solución final al problema de Riemman en términos de las variables originales

$$\mathbf{U}(x, t) = \sum_{i=I+1}^m \alpha_i \mathbf{K}^{(i)} + \sum_{i=1}^I \beta_i \mathbf{K}^{(i)}$$

donde el entero  $I = I(x, t)$  es el máximo valor de los sub índices  $i$  para el cual  $x - \lambda_i t > 0$ .

### A.3. Ley de conservación

Prestemos nuestra atención a los elementos esenciales para las aplicaciones numéricas de las leyes de conservación. Veremos en esta sección que restringimos a un problema más simple. Hemos definido anteriormente un sistema  $m$  dimensional de las leyes conservativas

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0$$

donde  $\mathbf{U}$  es el vector de variables conservativas y  $\mathbf{F}(\mathbf{U})$  es el vector flujo. Este sistema es hiperbólico si la matriz Jacobina

$$\mathbf{A}(\mathbf{U}) = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$$

tiene los valores propios  $\lambda_i(\mathbf{U})$  reales y tienen un sistema completo de vectores propios linealmente independientes  $\mathbf{K}^{(i)}(\mathbf{U})$ ,  $i = 1, \dots, m$ , los cuales pueden ser ordenados del siguiente modo

$$\lambda_1(\mathbf{U}) < \lambda_2(\mathbf{U}) < \dots < \lambda_m(\mathbf{U}),$$

$$\mathbf{K}^{(1)}(\mathbf{U}), \mathbf{K}^{(2)}(\mathbf{U}), \dots, \mathbf{K}^{(m)}(\mathbf{U}).$$

#### A.3.1. La forma integral de la ley de conservación

Existen dos buenas razones para considerar la forma integral de la ley de conservación

- i) La obtención de las ecuaciones está basada en los principios de conservación físico expresado con relaciones integrales sobre volúmenes de control,
- ii) La formulación integral requiere menos suavidad de las soluciones, por lo que podemos extender la solución a una clase de soluciones admisibles que incluye soluciones discontinuas.

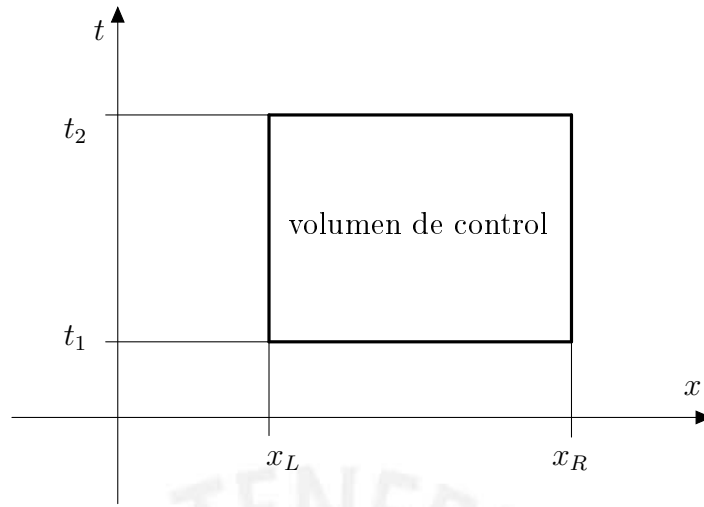


Figura A.2: Volumen de control  $V = [x_L, x_R] \times [t_1, t_2]$  sobre el plano  $x - t$ .

Consideremos el sistema dependiente unidimensional de la ecuación de Euler, cuya ecuación sobre el volumen de control  $V = [x_L, x_R] \times [t_1, t_2]$  sobre el plano  $x - t$  mostrado en la Figura A.2.

La ecuación de conservación de masa en el espacio unidimensional es

$$\frac{d}{dt} \int_{x_L}^{x_R} \rho(x, t) dx = f(x_L, t) - f(x_R, t),$$

donde  $f = \rho u$  es el flujo. Para el sistema completo tenemos

$$\frac{d}{dt} \int_{x_L}^{x_R} \mathbf{U}(x, t) dx + \mathbf{F}(\mathbf{U}(x_R, t)) - \mathbf{F}(\mathbf{U}(x_L, t)) = 0,$$

donde  $\mathbf{F}(\mathbf{U})$  es el vector flujo. Esta es una versión de la forma integral de la ley de conservación.



# Apéndice B

## Subprogramas

### Elementos de la programación simbólica en Python

#### Librerías

```
1 import numpy as np
2 import numpy.linalg
3 import matplotlib
4 import matplotlib.pyplot as plt
5 from numpy import zeros, ones
6 from matplotlib import pyplot as plt
7 from numpy import array, arange
8 import scipy.integrate as integrate
9 import scipy.special as special
10 from sympy import *
11 import fractions from fractions import Fraction
```

#### Solver para calcular simbólicamente un SEL tridiagonal cíclico por bloques

```
1 def TriBlockSolverCicl(a, b, c, d):
2     '''
3     TRIBLOCKSOLVERCICL: Resuelve un sistema tridiagonal en
4     bloques cíclico
```

```

4   para el cálculo de los coeficientes de reconstrucción
      considerando condiciones
5   de frontera periódicas.
6   A = [ b  c ]   a, b,c, d: Lista de matrices en bloque de
      un sistema A x = d
7       [ a  b ]
8   ',,'
9   ac, bc, cc, fc = map(np.array, (a, b, c, d))      # copia los
      arreglos
10  nf = len(bc)      # número de ecuaciones
11  alpha = 1
12  gamma = 1
13  bc[0] = bc[0]- fractions.Fraction(gamma,alpha)*cc[nf-1]
14  bc[nf-1]= bc[nf-1]-fractions.Fraction(gamma,alpha)*ac[0]
15  T=myinverse(bc[0])
16  cc[0] = np.dot(T,cc[0])
17  fc[0] = np.dot(T,fc[0])
18  bc[0] = T /alpha
19
20  for k in range(1,nf):
21      T      = myinverse(bc[k] - np.dot(ac[k] ,cc[k-1]))
22      fc[k] = np.dot(T, fc[k]-np.dot(ac[k],fc[k-1])) )
23      bc[k] = -np.dot(T,np.dot(ac[k] ,bc[k-1]))
24      if k < nf-1:
25          cc[k]=np.dot(T, cc[k])
26
27  bc[nf-1] = T/gamma + bc[nf-1]
28  for k in range(nf-2,-1,-1):
29      fc[k] = fc[k]-np.dot(cc[k],fc[k+1])
30      bc[k] = bc[k]-np.dot(cc[k],bc[k+1])
31
32  kk = len(ac[0]) # muestra error en caso kk=1 si es que se
      tiene una identidad simbólica
33  if kk==1 :
34      u= np.dot( myinverse(np.eye(kk)+alpha*np.dot(ac[0],bc[nf
          -1])+gamma*np.dot(cc[nf-1],bc[0])),alpha*np.dot(ac[0] ,
          fc[nf-1])+gamma*np.dot(cc[nf-1],fc[0])) )
35  else :

```

```

36     u= np.dot( myinverse(Identity(kk)+alpha*np.dot(ac[0],bc[
        nf-1]))+gamma*np.dot(cc[nf-1],bc[0])),alpha*np.dot(ac
        [0],fc[nf-1])+gamma*np.dot(cc[nf-1],fc[0]) )
37
38     for k in range(0,nf):
39         fc[k] = fc[k]-np.dot(bc[k],u)
40
41     del bc, cc, ac
42
43     return fc

```

## Definición de parámetros y número de onda

```

1 x = symbols('x', real=True)
2 K = symbols('K', real=True)
3 w1 = symbols('w1', real=True)
4 w2 = symbols('w2', real=True)
5 w3 = symbols('w3', real=True)
6 w4 = symbols('w4', real=True)
7 w5 = symbols('w5', real=True)
8 w6 = symbols('w6', real=True)
9 w7 = symbols('w7', real=True)
10 w=[1,w1, w2,w3,w4,w5,w6,w7]

```

## Cálculo de matrices de reconstrucción

```

1
2     A1 = 0.0*numpy.ndarray((M+1,k)) #A_{i-1}^{i}
3     B1 = 0.0*numpy.ndarray((M+1,k)) #B_{i-1}^{i}
4     A2 = 0.0*numpy.ndarray((M+1,k)) #A_{i+1}^{i}
5     B2 = 0.0*numpy.ndarray((M+1,k)) #B_{i+1}^{i}
6     bm1 = 0.0*numpy.ndarray((M+1,1)) #b_{i-1}^{i}
7     bm2 = 0.0*numpy.ndarray((M+1,1)) #b_{i+1}^{i}
8     D = 0.0*numpy.ndarray((N,k,k))
9     E = 0.0*numpy.ndarray((N,k,k))
10    F = 0.0*numpy.ndarray((N,k,k))
11    G = 0.0*numpy.ndarray((N,k,1))

```

```

12     ui = 0.0*numpy.ndarray((N,k,1))
13     for i in range(0,N):# == i=1,2,...,N-1,N    eq (12)
14         if i != 0:
15             j=i-1
16             dxj=xv[j+1]-xv[j]; dxi=xv[i+1]-xv[i];xvbi=xvb[i];
17             xvbj=xvb[j]
18             for m in range(0,M+1): # m=0,... M
19                 for l in range(0,k): # l=0,...,k-1 (l = 1,...k)
20                     A1[m,l]= (om(i,m,dxi)/dxj)*integrate(diff(
21                         varphi(x,l+1,dxi,xvbi),x,m),(x, xv[j],xv[j
22                         +1]))
23                     B1[m,l]= (om(i,m,dxi)/dxj)*integrate(diff(
24                         varphi(x,l+1,dxj,xvbj),x,m),(x, xv[j],xv[j
25                         +1]))
26                     bm1[m]=dirco(m)*(ubar[j]-ubar[i])
27         else:
28             i=0
29             j=i-1
30             dxj=xv[N]-xv[N-1]; dxi=xv[i+1]-xv[i];xvbi=xvb[i];
31             xvbj=xv[i]-dxj/2.0
32             for m in range(0,M+1): # m=0,... M
33                 for l in range(0,k): # l=0,...,k-1 (l = 1,...k)
34                     A1[m,l]= (om(i,m,dxi)/dxj)*integrate(diff(
35                         varphi(x,l+1,dxi,xvbi),x,m),(x, xv[i]-dxj,
36                         xv[j+1]))
37                     B1[m,l]= (om(i,m,dxi)/dxj)*integrate(diff(
38                         varphi(x,l+1,dxj,xvbj),x,m),(x, xv[i]-dxj,
39                         xv[j+1]))
40                     bm1[m]=dirco(m)*(ubar[j]-ubar[i])
41         if i != N-1:
42             j=i+1
43             dxj=xv[j+1]-xv[j]; dxi=xv[i+1]-xv[i];xvbi=xvb[i];
44             xvbj=xvb[j]
45             for m in range(0,M+1): # m=0,... M
46                 for l in range(0,k): # l = 0,...,k-1
47                     A2[m,l]= (om(i,m,dxi)/dxj)*integrate(diff(
48                         varphi(x,l+1,dxi,xvbi),x,m),(x, xv[j],xv[j
49                         +1]))

```

```

38         B2[m,1]= (om(i,m,dxi)/dxj)*integrate(diff(
           varphi(x,l+1,dxj,xvbj),x,m),(x, xv[j],xv[j
           +1]))
39         bm2[m]=dirco(m)*(ubar[j]-ubar[i])
40     else:
41         i=N-1
42         j=i+1
43         dxj=xv[1]-xv[0]; dxi=xv[i+1]-xv[i]; xvbj=xvb[i];
           xvbj=xv[i+1]+dxj/2.0
44     for m in range(0,M+1): # m=0,... M
45         for l in range(0,k): # l=0,...,k-1 (l = 1,...,k)
46             A2[m,1]= (om(i,m,dxi)/dxj)*integrate(diff(
           varphi(x,l+1,dxi,xvbi),x,m),(x, xv[j],xv[j
           ]+dxj))
47             B2[m,1]= (om(i,m,dxi)/dxj)*integrate(diff(
           varphi(x,l+1,dxj,xvbj),x,m),(x, xv[j],xv[j
           ]+dxj))
48             bm2[m]=dirco(m)*(ubar[0]-ubar[i])
49
50
51     D[i] = np.dot(A1.transpose(),A1)+ np.dot(A2.transpose()
           ,A2)
52     E[i] = -np.dot(A1.transpose(),B1)
53     F[i] = -np.dot(A2.transpose(),B2)
54     G[i] = np.dot(A1.transpose(), bm1) + np.dot(A2.transpose
           (), bm2)
55
56     ui=TriBlockSolverCicl(E, D, F, G)
57     return ui

```

## Cálculo del número de onda modificado

```

1     ui = TriBlockSolverCicl(E, D, F, G)
2     j = 1
3     uLimeum = eval_reconstrucfunc_int_i(ubar[j],xv,xvb,ui,k,xv[j
           +1],j)
4     fimeum = f(uLimeum)
5     j = j + 1

```

```

6     uLimeum = eval_reconstrucfunc_int_i(ubar[j],xv,xvb,ui,k,xv[j
      +1],j)
7     fimaum = f(uLimeum)
8     flux_num = fimaum - fimeum
9     flux_exc = exp(2*K*I)-exp(K*I)
10    kp = (flux_num/flux_exc)*K
11    re_part=re(expand(simplify(kp),complex=True))
12    im_part=im(expand(simplify(kp),complex=True))

```

## Cabeceras de subprogramas

```

1 def TriblockSolverv1(a, b, c, d):
2     '''
3     TRIBLOCKSOLVERV: Resuelve un sistema de ecuaciones
      tridiagonales usado para el cálculo
4     de las coeficientes de reconstrucción considerando
      condiciones de frontera de Dirichlet
5     A = [ b  c ]   a, b,c, d: Lista de matrices en bloque de
      un sistema A x = d
6         [ a  b ]
7 def TriblockSolverCicl(a, b, c, d):
8     '''
9     TRIBLOCKSOLVERCICL: Resuelve un sistema tridiagonal en
      bloques cíclico
10    para el cálculo de los coeficientes de reconstrucción
      considerando condiciones
11    de frontera periódicas.
12    A = [ b  c ]   a, b,c, d: Lista de matrices en bloque de
      un sistema A x = d
13        [ a  b ]
14
15 def evalreconstrucfunc(ubar,xv,xvb,ui,k,x): # Es lo mismo que
      evalSplineCom
16     '''
17     EVALRECONSTRUCFUNC: Evalua el polinomio de reconstrucción
      de orden k
18     Input: ubar: valores promedios
19           xv : extremos de intervalos

```

```

20         xvb : puntos medios de los intervalos
21         ui  : coeficientes de reconstrucción
22         k   : orden del polinomio de reconstrucción
23     Output: s : evaluación del polinomio de reconstrucción
24
25 def  coefreconstrucdirichlet(ubar,xv,xvb, k,M,N):
26     '''
27     COEFRECONSTRUCDIRICHLET: Calcula las matrices de
28         reconstrucción y los coeficientes del polinomio de
29         reconstrucción. para condiciones de frontera de Dirichlet.
30     Input: ubar: valores promedios
31            xv  : extremos de intervalos
32            xvb : puntos medios de los intervalos
33            k   : orden del polinomio de reconstrucción
34            M   : Parámetro del método ( siempre lo considero M=
35                  k-1)
36            N   : tamaño de partición
37     Output: ui  : coeficientes de reconstrucción
38     '''
39
40 def  coefreconstrucperiod(ubar,xv,xvb, k,M,N):
41     '''
42     COEFRECONSTRUCPERIOD: Calcula las matrices de
43         reconstrucción y los coeficientes del polinomio de
44         reconstrucción para condiciones de frontera periódicas
45     Input: ubar: valores promedios
46            xv  : extremos de intervalos
47            xvb : puntos medios de los intervalos
48            k   : orden del polinomio de reconstrucción
49            M   : Parámetro del método ( siempre lo considero M=
50                  k-1)
51            N   : tamaño de partición
52     Output: ui  : coeficientes de reconstrucción
53     '''
54
55 def  solveredpdir(ubar,xv,xvb, k,M,N,cfl,RK):
56     '''
57     SOLVEREDPDIR itera en el tiempo usando métodos de RK y
58         calculando los flujos

```

```

54     numéricos para condiciones de frontera de Dirichlet
55     Input: ubar: valores promedios
56           xv  : extremos de intervalos
57           xvb : puntos medios de los intervalos
58           k   : orden del polinomio de reconstrucción
59           M   : Parámetro del método ( siempre lo considero M=
                k-1)
60           N   : tamaño de partición
61           cfl : CFL
62           RK  : Método de RK (2, 3 o 4)
63     Output: ubar : Valores promedios luego de la iteración
                temporal
64     '''
65
66     main()
67         for i in range(0,N): # i= 0,...,N-1
68             ubar[i]=integrate(u(x),(x,xv[i],xv[i+1]))/(xv[i+1]-
                xv[i])
69             # -----
70             if condfr==1:
71                 ubar=solveredpper(ubar,xv,xvb, k,M,N,cfl,RK)
72             elif condfr==2:
73                 ubar=solveredpdir(ubar,xv,xvb, k,M,N,cfl,RK)
74             # -----
75     Solución aproximada
76     z[i]=evalreconstrucfuncinti(ubar[i],xv,xvb,ui,k,xv[i],i
        )

```



# Bibliografía

- [1] Ernest Bladé Castellet. *Modelación del flujo en lámina libre sobre cauces naturales. Análisis integrado con esquemas en volúmenes finitos en una y dos dimensiones*. PhD thesis, Universitat Politècnica de Catalunya. Departament d'Enginyeria Hidràulica, Marítima i Ambiental, 01 2005.
- [2] Jiri Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.
- [3] J. E: Casavilca. *p-Multigrid explícito para um método de volumes finitos de alta-ordem nao estruturado*. PhD thesis, Instituto de Matemática e Estatística. University of Sao Paulo. Sao Paulo. Retrieved 2016/11/10, 2016.
- [4] Erwin Fehlberg. *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*. NASA, 1969.
- [5] Oliver Friedrich. Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids. *Journal of Computational Physics*, 144(1):194 – 212, 1998.
- [6] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. *Math. Comput.*, 67(221):73–85, January 1998.
- [7] Antony Jameson, Wolfgang Schmidt, Eli Turkel, et al. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA paper*, 1259:1981, 1981.
- [8] M. P. Martín, E. M. Taylor, M. Wu, and V. G. Weirs. A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence. *Journal of Computational Physics*, 220:270–289, December 2006.

- [9] K Jarrod Millman and Michael Aivazis. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.
- [10] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*. Fluid Mechanics and Its Applications. Springer International Publishing, 2015.
- [11] Carl Ollivier-Gooch, Amir Nejat, and Krzysztof Michalak. Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations. *AIAA journal*, 47(9):2105–2120, 2009.
- [12] Carl F Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.
- [13] J. Peraire and P.-O. Persson. The Compact Discontinuous Galerkin (CDG) Method for Elliptic Problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.
- [14] B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*. Society for Industrial and Applied Mathematics, 2008.
- [15] Steven J Ruuth and Raymond J Spiteri. High-order strong-stability-preserving runge–kutta methods with downwind-biased spatial discretizations. *SIAM Journal on Numerical Analysis*, 42(3):974–996, 2004.
- [16] Chi-Wang Shu. Total-variation-diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084, 1988.
- [17] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced numerical approximation of nonlinear hyperbolic equations*, pages 325–432. Springer, 1998.
- [18] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, 2004.

- [19] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 3rd edition, 2009.
- [20] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [21] R. Vichnevetsky. *Fourier Analysis of Numerical Approximations of Hyperbolic Eqns*. SIAM, 1982.
- [22] Qian Wang, Yu-Xin Ren, and Wanai Li. Compact high order finite volume method on unstructured grids I: Basic formulations and one-dimensional schemes. *Journal of Computational Physics*, pages –, 2016.
- [23] Z.J. Wang. Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids. Basic Formulation. *Journal of Computational Physics*, 178(1):210 – 251, 2002.
- [24] Z.J. Wang and Yen Liu. Spectral (finite) volume method for conservation laws on unstructured grids. *Journal of Computational Physics*, 179(2):665 – 697, 2002.
- [25] Z.J Wang, Laiping Zhang, and Yen Liu. Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems. *Journal of Computational Physics*, 194(2):716 – 741, 2004.
- [26] W. Echegaray X. Chávez. Métodos multimalla para la solución de ecuaciones diferenciales parciales elípticas anisotrópicas. *Revcuini, Revista de la Facultad de Ciencias de la UNI*, 12:1–19, 2007.
- [27] Laiping Zhang, Liu Wei, He Lixin, Deng Xiaogang, and Zhang Hanxin. A class of hybrid DG/FV methods for conservation laws I: Basic formulation and one-dimensional systems. *Journal of Computational Physics*, 231(4):1081 – 1103, 2012.